Creating and Using Semantics for Information Retrieval and Filtering

State of the Art and Future Research

Several experiments have been carried out in the last 15 years investigating the use of various resources and techniques (e.g., thesauri, synonyms, word sense disambiguation, etc.) to help refine or enhance queries. However, the conclusions drawn on the basis of these experiments vary widely. Results of some studies have led to the conclusion that semantic information serves no purpose and even degrades results, while others have concluded that the use of semantic information drawn from external resources significantly increases the performance of retrieval software. At this point, several question arise:

- Why do these conclusions vary so widely?
- Is the divergence a result of differences in methodology?
- Is the divergence a result of a difference in resources? What are the most suitable resources?
- Do results using manually constructed resources differ in significant ways from results using automatically extracted information?
- From corpus building to terminology structuring, to which methodological requirements resources acquisition has to comply with in order to be relevant to a given application?
- What is the contribution of specialized resources?
- Are present frameworks for evaluation (e.g., TREC) appropriate
- for evaluation of results?.

These questions are fundamental not only to research in document retrieval, but also for information searching, question answering, filtering, etc. Their importance is even more acute for multilingual applications, where, for instance, the question of whether to disambiguate before translating is fundamental.

Moreover, the increasing diversity of monolingual as well as multilingual documents on the Web invite to focus attention on lexical variability in connection with textual genre and with questioning the resources reusability stance.

The goal of this workshop is to bring together researchers in the domain of document retrieval, and in particular, researchers on both sides of the question of the utility of enhancing queries with semantic information gleaned from languages resources and processes.

The workshop will provide a forum for presentation of the different points of view, followed by a roundtable in which the participants will assess the state of the art, consider the results of past and on-going work and the possible reasons for the considerable differences in their conclusions. Ultimately, they will attempt to identify future directions for research.

Workshop Organisers

Christian Fluhr, CEA, France
Nancy Ide, Vassar College, USA
Claude de Loupy, Sinequa, France
Adeline Nazarenko, LIPN, Université de Paris-Nord, France
Monique Slodzian, CRIM, INALCO, France

Workshop Programme Committee

Table of Contents

Finding Similar Documents in Document Collections

Thorsten Brants and Reinhard Stolle

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd, Palo Alto, CA 94304, USA
{brants,stolle}@parc.com

Abstract

Finding similar documents in natural language document collections is a difficult task that requires general and domain-specific world knowledge, deep analysis of the documents, and inference. However, a large portion of the pairs of similar documents can be identified by simpler, purely word-based methods. We show the use of Probabilistic Latent Semantic Analysis for finding similar documents. We evaluate our system on a collection of photocopier repair tips. Among the 100 top-ranked pairs, 88 are true positives. A manual analysis of the 12 false positives suggests the use of more semantic information in the retrieval model.

1. Introduction

Collections of natural language documents that are focused on a particular subject domain are commonly used by communities of practice in order to capture and share knowledge. Examples of such "focused document collections" are FAQs, bug-report repositories, and lessons-learned systems. As such systems become larger and larger, their authors, users and maintainers increasingly need tools to perform their tasks, such as browsing, searching, manipulating, analyzing and managing the collection. In particular, the document collections become unwieldy and ultimately unusable if obsolete and redundant content is not continually identified and removed.

We are working with such a knowledge-sharing system, focused on the repair of photocopiers. It now contains about 40,000 technician-authored free text documents, in the form of tips on issues not covered in the official manuals. Such systems usually support a number of tasks that help maintain the utility and quality of the document collection. Simple tools, such as keyword search, for example, can be extremely useful. Eventually, however, we would like to provide a suite of tools that support a variety of tasks, ranging from simple keyword search to more elaborate tasks such as the identification of "duplicates." Fig. 1 shows a pair of similar tips from our corpus. These two tips are about the same problem, and they give a similar analysis as to why the problem occurs. However, they suggest different solutions: Tip 118 is the "official" solution, whereas Tip 57 suggests a short-term "work-around" fix to the problem. This example illustrates that "similarity" is a complicated notion that cannot always be measured along a one-dimensional scale. Whether two or more documents should be considered "redundant" critically depends on the task at hand. In the example of Fig. 1, the work-around tip may seem redundant and obsolete to a technician who has the official new safety cable available. In the absence of this official part, however, the work-around tip may be a crucial piece of information.

Our goal is to develop techniques that analyze the conceptual contents of natural language documents at a granularity that is fine enough to capture distinctions like the one between Tips 57 and 118, described in the previous paragraph. In order to do that, we are designing formal representations of document contents that will allow us to assess not only whether two documents are about the same subject but also whether two documents actually say the same thing. We are currently focusing on the tasks of computer-assisted redundancy resolution. We hope that our techniques will eventually extend to support even more ambitious tasks such as the identification and resolution of inconsistent knowledge, knowledge fusion, question answering, and trend analysis.

We believe that, in general, the automated or computer-assisted management of collections of natural language documents requires a fine-grained analysis and representation of the documents' contents. This fine granularity in turn mandates deep linguistic processing of the text and inference capabilities using extensive linguistic and world knowledge. Following this approach, our larger research group has implemented a prototype, which we will briefly describe in the next section. This research prototype system is far from complete. Meanwhile, we are investigating to what extent currently operational techniques are useful to support at least some of the tasks that arise from the maintenance of focused document collections. We have investigated the utility of Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999b) for the task of finding similar documents. Section 3. describes our PLSA model and Section 4. reports on our experimental results in the context of our corpus of repair tips. In that section, we also attempt to characterize the types of similarities that are easily detected and contrast them to the types that are easily missed by the PLSA technique. Finally, we speculate how symbolic knowledge representation and inference techniques that rely on a deep linguistic analysis of the documents may be coupled with statistical techniques in order to improve the results.

2. Knowledge-Based Approach

Our goal is to build a system that supports a wide range of knowledge management tasks for focused document collections. We believe that powerful tools for tasks like redundancy resolution, topic browsing, question answering, knowledge fusion, and so on, need to analyze and represent the documents' conceptual contents at a fine level of granularity.

Concentrating on the task of redundancy resolution, our

	Tip 57		Tip 118
Problem:	Left cover damage	Problem:	The current safety cable used in the 5100 Document Handler fails prematurely, causing the Left Document Handler Cover to break.
Cause:	The left cover safety cable is breaking, allowing the left cover to pivot too far, breaking the cover.	Cause:	The plastic jacket made the cable too stiff. This causes stress to be concentrated on the cable ends, where it eventually snaps.
Solution:	Remove the plastic sleeve from around the cable. Cutting the plastic off of the cable makes the cable more flexible, which prevents cable breakage. Cable breakage is a major source of damage to the left cover.	Solution:	When the old safety cable fails, replace it with the new one, which has the plastic jacket shortened.

Figure 1: Example of Eureka tips

project group has so far built a prototype whose goal is to identify conceptually similar documents, regardless of how they are written. This task requires extensive knowledge about language and of the world. Since most of this knowledge engineering effort is performed by hand at the moment, our system's coverage is currently limited to fifteen pairs of similar tips. We are in the process of scaling the system up by one to two orders of magnitude. Eventually, we hope to also support more general tasks, namely identify the parts of two documents that overlap; and identify parts of the documents that stand in some relation to each other, such as expanding on a particular topic or being in mutual contradiction. Such a system will enable the maintenance of vast document collections by identifying potential redundancies or inconsistencies for human attention.

State-of-the-art question answering and information extraction techniques (e.g., (Bear et al., 1997)) are sometimes able to identify entities and the relations between them at a fine level of granularity. However, the functionality and coverage of these techniques is typically restricted to a limited set of types of entities and relations that have been formalized upfront using static templates. Like a small number of other research projects (e.g., the TACITUS project (Hobbs et al., 1993)), our approach is based on the belief that the key to solving this problem is a principled technique for producing formal representations of the conceptual contents of the natural language documents. In our approach, a deep analysis based on Lexical Functional Grammar theory (Kaplan and Bresnan, 1982) combined with Glue Semantics (Dalrymple, 1999) produces a compact representation of the syntactic and semantic structures for each sentence. From this language-driven representation of the text, we map to a knowledge-driven representation of the contents that abstracts away from the particular natural language expression. This mapping includes several—not necessarily sequential—steps. In one step, we rely on a domain-specific ontology to identify canonicalized entities and events that are talked about in the text. In our case, these entities and events include things like parts, e.g., photoreceptor belt, and relevant activities such as cleaning, for example. Another step performs thematic role assignments and assembles fragments of conceptual structures from the normalized entities and events (e.g., cleaning a photoreceptor belt). Furthermore, certain relations are normalized; for example, "stiff" and "flexible" (in Fig. 1) both refer to the rigidity of an object, one being the inverse of the other. Yet another step composes structure fragments into higher-level structures that reflect causal or temporal relations, such as action sequences or repair plans. All steps involve ambiguity resolution as a central problem, which requires inference based on extensive linguistic and world knowledge. For a more detailed description of this approach and its scalability, see (Crouch et al., 2002).

Finally, we assess the similarity of two documents using a variant of the Structure Mapping Engine (SME) (Forbus et al., 1989). SME anchors its matching process in identical elements that occur in the same structural positions in the base and target representations, and from this builds a correspondence. The larger the structure that can be recursively constructed in this manner, while preserving a systematicity constraint of one-to-one correspondence between base and target elements and the identicality of anchors, the greater the similarity score.

We expect that the fine-grained conceptual representations discussed in this section will eventually enable our system to detect whether two documents are not only about the same subject but also saying the same thing. Many interesting cases of similarity can, however, be detected with lighter-weight techniques. This is the topic of the next section.

3. The Word-Based Statistical Model

While in the general case deep processing, knowledge about the world, and inference are necessary to identify similar documents, there may be a large number of similar pair that can be discovered by a shallow approach. We now view the task of finding similar pairs of documents as an information retrieval problem where documents are matched based on the words that occur in the documents, i.e., we use a vector space model of the documents. Comparison is done using Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999b).

3.1. Document Preprocessing

Each document is first preprocessed by:

1. Separating the document fields. Each tip usually comes with additional administrative information like author, submission date, location, status, contact information, etc. We extract the information that is contained in the `CHAINS`, `PROBLEM`, `CAUSE`, and `SO-`

LUTION fields[1].

2. Tokenizing the document. Words and numbers are separated at white space, punctuation is stripped, abbreviations are recognized.

3. Lemmatizing each token, i.e., each word is uniquely mapped to a base form. We use the LinguistX lemmatizer[2] to perform this task.

Steps 1 to 3 identify the terms in the vocabulary. We select the subset of those terms that occur in at least two documents. Given this vocabulary, each document d is represented by its term-frequency vector $f(d,w)$, where w are the terms of the document.

3.2. Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Analysis (PLSA) is a statistical latent class model or aspect model (Hofmann, 1999a; Hofmann, 1999b). It can be seen as a statistical view of Latent Semantic Analysis (LSA) (Deerwester et al., 1990). The model is fitted to a training corpus by the Expectation Maximization (EM) algorithm (Dempster et al., 1977). It assigns probability distributions over classes to words and documents and thereby allows them to belong to more than one class, and not to only one class as is true of most other classification methods. PLSA represents the joint probability of a document d and a word w based on a latent class variable z:[3]

$$P(d,w) = P(d)\sum_z P(w|z)P(z|d) \tag{1}$$

The model makes an independence assumption between word w and document d if the latent class z is given, i.e., $P(w|z,d) = P(w|z)$. PLSA has the following view of how a document is generated: first a document $d \in \mathcal{D}$ (i.e., its dummy label) is chosen with probability $P(d)$. For each word in document d, a latent topic $z \in \mathcal{Z}$ is chosen with probability $P(z|d)$, which in turn is used to choose the word $w \in \mathcal{W}$ with probability $P(w|z)$.

A model is fitted to a document collection $\mathcal{D}$ by maximizing the log-likelihood function $\mathcal{L}$:

$$\mathcal{L} = \sum_{d\in\mathcal{D}}\sum_{w\in d} f(d,w) log P(d,w) \tag{2}$$

The E-step in the EM-algorithm is

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')} \tag{3}$$

and the M-step consists of

$$P(w|z) = \frac{\sum_d f(d,w)P(z|d,w)}{\sum_{d,w'} f(d,w')P(z|d,w')} \tag{4}$$

$$P(d|z) = \frac{\sum_w f(d,w)P(z|d,w)}{\sum_{d',w} f(d',w)P(z|d',w)} \tag{5}$$

$$P(z) = \frac{\sum_{d,w} f(d,w)P(z|d,w)}{\sum_{d,w} f(d,w)} \tag{6}$$

The parameters are either randomly initialized or according to some prior knowledge.

After having calculated the reduced dimensional representations of documents in the collection, we map the vectors back to the original term space to yield vectors $P(w|d)$ by

$$P(w|d) = \sum_z P(w|z)P(z|d) \tag{7}$$

$P(w|d)$ can be seen as a smoothed version of the empirical distribution $r(w|d) = f(d,w)/f(d)$ of words in the document. The advantage of the smoothed version is that it captures semantic similarities through the lower-dimensional representation.

Note that this process is intended for the pairwise comparison of all documents in the training collection. It can be extended to new documents q (query or test documents) by using the folding-in process. Folding-in uses Expectation-Maximization as in the training process; the E-step is identical, the M-step keeps all the $P(w|z)$ constant and recalculates $P_{fi}(z|q)$. Usually, a very small number of iterations is sufficient for folding-in. We get a smoothed representation of a folded-in document by

$$P_{fi}(w|q) = \sum_z P(w|z)P_{fi}(z|q) \tag{8}$$

This corresponds to the PLSI-U model described in (Hofmann, 1999b).

3.3. Document Comparison

A standard way of comparing vector space representations of documents d_1 and d_2 is to calculate the cosine similarity score of tf-idf weighted document vectors (Salton, 1988):

$$\mathrm{sim}_{\cos}(d_1,d_2) = \frac{\sum_w \hat{f}(d_1,w)\hat{f}(d_2,w)}{\sqrt{\sum_w \hat{f}(d_1,w)^2}\sqrt{\sum_w \hat{f}(d_2,w)^2}} \tag{9}$$

$\hat{f}(d,w)$ is the weighted frequency of word w in document d:

$$\hat{f}(d,w) = f(d,w)\log\frac{N}{df(w)} \tag{10}$$

where N is the total number of documents, and $df(w)$ is the number of documents containing word w.

We additionally perform the comparison on the PLSA representation of $P(w|d)$. Pairwise comparisons are done by

$$\mathrm{sim}^{\mathrm{PLSA}}_{\cos}(d_1,d_2) = \frac{\sum_w P(w|d_1)P(w|d_2)}{\sqrt{\sum_w P(w|d_1)^2}\sqrt{\sum_w P(w|d_2)^2}} \tag{11}$$

[1]The CHAINS field contains a numerical identifier of the product line.

[2]For information about the LinguistX tools, see www.inxight.com/products/linguistx/

[3]Unless otherwise noted, we use the following notational conventions: training documents $d, d' \in \mathcal{D}$, test documents $q, q' \in \mathcal{Q}$, words $w, w' \in \mathcal{W}$, and classes $z, z' \in \mathcal{Z}$.

Table 1: Precision of the statistical model for the n top-ranked pairs. A pair of tips is considered a "true positive" if their conceptual contents are categorized to be the same, similar, or in the subset relationship.

n	precision
10	100%
20	100%
30	100%
40	96%
50	92%
60	92%
70	90%
80	87%
90	88%
100	88%

Both similarities are combined with a weight λ to yield the final similarity score (see (Hofmann, 1999b)).

$$\mathrm{sim}(d_1, d_2) = \lambda \mathrm{sim}_{\cos}(d_1, d_2) + (1 - \lambda)\mathrm{sim}_{\cos}^{\mathrm{PLSA}}(d_1, d_2) \quad (12)$$

The output of the algorithm is a list of pairs ranked according to their similarity.

4. Experiments

We applied the algorithm described in Section 3. to a subset of the Eureka database consisting of 1,321 tips. PLSA representations of $P(w|d)$ were created for each tip, and pairs of tips were ranked according to their similarity. Following (Hofmann, 1999b), we created models with $Z = 32, 48, 64, 80, 128$ latent classes, calculated the average $P(w|d)$. The similarity score was combined with the standard tf-idf cosine similarity with a weight of $\lambda = \frac{1}{6}$.

4.1. Precision and Recall

We manually inspected the 100 top-ranked pairs of tips and classified their similarity by hand according to the types of similarity described in Section 4.2.. The results are shown in Table 1. Of the 10 top-ranked pairs, all 10 were actual duplicates,[4] of the 40 top-ranked pairs, 96% were true positives, and so on. The manual inspection of the 100 top-ranked pairs (of the potential 871,860 pairs) revealed 88 true positives.

Independent manual sampling of the subset of 1,321 tips, which is a very tedious and time-consuming task, revealed 17 similar pairs (14 pairs and 1 triple). 3 of these pairs were among the top 100 emitted by the word-based statistical model. This is a recall of 18% on the manually identified similar pairs. However, it is unclear how this number relates to the overall recall because the distribution of the other similar pairs is currently unclear.

[4] A pair of tips is considered "duplicates" if their conceptual contents are categorized to be the same. A pair of tips is considered a "true positive" if their conceptual contents are categorized to be the same, similar, or in the subset relationship. See Section 4.2..

Table 2: Number of pairs with structural and conceptual match in the 100 top-ranked pairs of documents. We are interested in finding the conceptually same/similar/subset pairs. False positives are shown in *italics*.

		conceptual				
		same	sim	subset	diff	sum
surface	same	24	0	10	*2*	36
	sim	17	24	13	*8*	62
	diff	0	0	0	*2*	2
	sum	41	24	23	*12*	100

4.2. Types of Similarity

The word-based statistical model of Section 3. seems to be good at identifying pairs whose texts are similar *at a surface level*. In order to see how well the model does at identifying pairs whose contents are *conceptually similar*, we manually performed a qualitative evaluation and classified each of the 100 top-ranked pairs according to the following criteria:

Surface similarity of texts: *same, similar, different*. Surface similarity describes the similarity of the set of words and syntactic constructions used in the documents. *Same* means that the documents are (almost) identical. *Similar* means that some words may be different or replaced by synonyms (e.g., "fault" vs. "failure" vs. "problem", "motor" vs. "drive", "line" vs. "wire", etc.), constructions are different, order of sentences may be different. *Different* means that the texts are different.

Conceptual similarity of contents: *same, similar, subset, different*. Conceptual similarity refers to the semantic/conceptual contents of the document, independent of how it is expressed as surface text. *Same* means that the documents have (almost) the same contents (e.g., "cutting the plastic off of the cable makes the cable more flexible" vs. "the plastic jacket made the cable too stiff"). *Similar* means that there is a significant overlap of conceptual contents between the two documents; for example, the tips describe the same problem but suggest different solutions (see Fig. 1), or, the tips describe an analogous problem exhibited at different mechanical parts (see Fig. 2).

Subset describes cases where the conceptual contents of one document form a proper subset of the conceptual contents of the other document—for example, if one document elaborates on the other. *Different* describes conceptually different documents.

Table 2 shows how many of the pairs fall into the different categories. Since the PSLA model is word-based, almost none of the pairs have different surface similarity. In the 100 top-ranked pairs, the majority of false positives occur when the surface texts are similar but the conceptual contents are different (8 out of 12).

The algorithm identifies surface similarity very well, only 2 out of 100 pairs are different at the surface text level.

Tip 690		Tip 714	
Problem:	08-110, Tray 3 misfeed	Problem:	08-100, Tray 1 misfeed
Cause:	J201 Pin 1 loose. Drive coupling set screw loose, Blower hose came off, Fang plate out of adjustment, Stack height out of adjustment, Defective DRCC1.	Cause:	Set screw on feed clutch loose. Stack height sensor out of bracket. Feeder drive coupling loose. Blower hose off.
Solution:	Reseat J201 Pin 1. Tighten drive coupling, Reconnect blower hose , Adjust fang plate, Adjust stack height. Replace DRCC1.	Solution:	Adjust clutch. Repair stack height sensor. Tighten feeder driv e coupling. Repair blower hose.

Figure 2: True positive: this pair at rank 68 has similar surface text and is similar at the conceptual level.

Tip 1280		Tip 1281	
Problem:	Xerox Binder 120. The "READY FOR AUTO FEED" message does not change when set clamp assy is pulled in	Problem:	Xerox Binder 120. The Binder 120 does not display "Ready for auto feed" message.
Cause:	Set Clamp extended sensor (Q23) is "H" all the time	Cause:	Set Clamp extended sensor (Q23) is "Lo" all the time
Solution:	check the set clamp sensor wires for an open circuit, if ok, Replace the set clamp extended sensor (Q23)	Solution:	Check the set clamp extended sensor wires for Short circuit to frame, Set clamp out flag is in the sensor correctly, if ok, replace the sensor.

Figure 3: False positive: this pair at rank 37 has almost the same surface text but is different at the conceptual level.

These two pairs involve very long documents (average of 1030 tokens per document compared to 132 tokens per document overall average). The documents have an overlap in vocabulary, but the sentences and sequences of sentences are very different.

Correlation with conceptual similarity can also be found, but it is smaller. 10 out of 100 pairs were categorized as the same or similar at the surface but are conceptually different; from the viewpoint of a user in the context of a conceptual task, these pairs should not be identified as similar tips. We believe that a deeper analysis of the document contents as outlined in Section 2. will help distinguish between conceptually different documents and, therefore, reduce the number of such false positives.

One of the two pairs that are almost the same at the surface level but have different conceptual contents is shown in Fig. 3.

They use the same or very similar words, but make opposite statements at the conceptual level. Tip 1280 describes a sensor signal that is erroneously "high" because of an open circuit. Tip 1281 describes a sensor signal that is erroneously "low" because of a short circuit. This difference cannot be found by the word-based statistical model. The topics of these two documents are very similar; however, a correct analysis of the contents requires the recognition of the difference between "does not display" and "does not change", the difference between "Lo" and "H", and the difference between "open circuit" and "short circut" despite the fact that these phrases often occur in similar contexts.

Fig. 4 shows a pair with similar surface texts but different conceptual contents. Tip 227 explains how to repair or prevent a particular failure that is caused by a ring's wearing out. Tip 173 says that an improved repair kit can be ordered; it also provides a work-around for the case in which that improved kit is not available.

The two examples in Figures 3 and 4 show that in many cases it is necessary to process the text more deeply than at the word level in order to be able to recognize fine-grained distinctions in the documents' contents. On the other hand, a large number of true positives are actually discovered by the word-based model (88 out of the 100 top-ranked pairs). The word-based statistical model even finds cases in which the conceptual contents are similar, but where this fact is not immediately obvious from the surface-level texts. Fig. 2 shows an example of this case. The two tips describe almost the same fault situation, except that one of them occurs in connection with Tray 1 while the other one occurs in connection with Tray 3. Even for a human—at least for an untrained human—, this pair is difficult to detect.

The examples suggests that symbolic and statistical techniques may be good at different tasks that complement each other nicely. Statistical techniques seem to be good at identifiying that the two tips are about the same topic. Knowledge-based techniques—specifically, a domain ontology—may help distinguish "Fuser Couplings" from the "Fuser Couplings and Shaft Repair Kit" (cf. Fig. 4), which in turn may trigger further distinctions between the two tips based on domain-specific knowledge. Similarly, the example in Fig. 3 suggests that a statistical analysis coupled with a limited normalization of relations that occur frequently in the domain may be a promising direction to pursue.

Fig. 5 shows the rank of a pair vs. its similarity. Our data set contains 1,321 documents, i.e., there are 871,860 pairs. Word-based similarity does not decrease linearly. There is a large drop at the beginning, then the curve is relatively flat, and it suddenly drops again at the very end. All of the manually found similar pairs (the 17 pairs described in Section 4.1.) are marked with a ∘ in the graph; they are among the first 7% (the lowest rank is 57,014). We do currently not

	Tip 173		Tip 227
Problem:	Improved Fuser Couplings 600K31031 Tag P-184. Broken calls when servicing failed Fuser Drive Couplings.	Problem:	Fuser Couplings and Shaft Repair Kit, 605K3950, Tag P-129. The retaining ring that holds the Fuser Assembly Drive Coupling in place wears out and falls off the shaft.
Cause:	The parts needed to repair a Fuser Drive failure are presently contained in two separate Kits. If the service representative does not have both Kits in inventory the service call is interrupted.	Cause:	The Fuser Assembly Drive Coupling rubs against the retaining ring as it turns.
Solution:	1. To repair Fuser Drive failures, order the new Fuser Couplings an d Shaft Repair Kit 600K31031, TAG P-184. This kit contains all the parts in Fuser Couplings and Shaft Repair Kit 605K3950 except that the improved Drive Coupling, issued separately in Kit 600K31030, has been substituted. 2. If you have 600K31030 as well as 605K3950 in inventory, these Kits can be salvaged to provide the same parts as the new Kit. Open 605K3950 and discard only the Fuser Drive Coupling, then use the Coupling contained in Kit number 600K31030 in its place.	Solution:	On the next service call check to see if P-129 is installed. If Tag P-129 is not installed, order and install the Fuser Couplings and Shaft Repair Kit, 605K3950.

Figure 4: False positive: this pair at rank 86 has similar surface text and is about similar parts, but is different at the conceptual level.

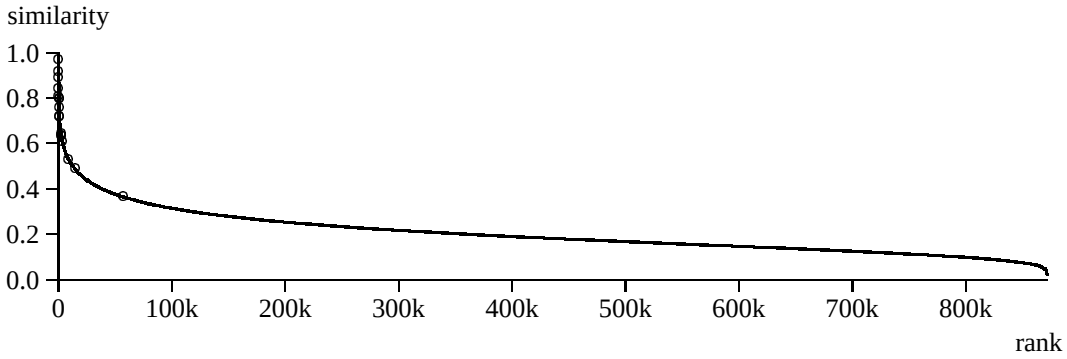


Figure 5: Rank vs. PLSA similarity. Manually found pairs are marked with ∘.

know whether there are any similar pairs below this rank, but it is probably safe to assume that almost all of the similar pairs are within the initial portion of the graph. Even if the presented statistical method does not rank all similar pairs at the very top, it seems to efficiently place them in a small initial segment at the top.

One focus of our current research effort is to understand the capabilities and limitations of the current PLSA model in order to design an improved system by, for example, (1) supplying the PLSA model with better-suited information for any given particular task, or (2) using the current version of the PLSA model as a prefilter for the knowledge-based approach.

5. Conclusions

We address the problem of matching the conceptual contents of documents. The domain of the documents in our experiments is the repair of photocopiers. In general, the problem requires world knowledge and deep processing of the documents. But in a large number of cases, similar documents can be found by shallow processing and a word-based statistical model. A quantitative evaluation shows that 88 of the 100 statistically top-ranked documents are true positives. An analysis of the erroneous cases indicates where the statistical model could benefit from deeper processing. Two important types of information that are currently absent from our statistical model are negation and relations between entities. We expect that providing the model with more semantic information along these lines will improve our system's performance and allow it to make finer distinctions among the documents' contents.

6. References

J. Bear, D. Israel, J. Petit, and D. Martin. 1997. Using information extraction to improve document retrieval. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, pages 367–377. NIST.

R. Crouch, C. Condoravdi, R. Stolle, T. King, V. de Paiva, J. O. Everett, and D. G. Bobrow. 2002. Scalability of redundancy detection in focused document collections. In *Proceedings First International Workshop on Scalable Natural Language Understanding (SCANALU-2002)*, Heidelberg, Germany.

M. Dalrymple, editor. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. MIT Press, Cambridge, MA.

S. Deerwester, S. Dumais, G. W. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–21.

K. D. Forbus, B. Falkenhainer, and D. Gentner. 1989. The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63.

J. R. Hobbs, M. Stickel, S. Appelt, and P. Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

T. Hofmann. 1999a. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, Sweden.

T. Hofmann. 1999b. Probabilistic latent semantic indexing. In *Proceedings of SIGIR-99*, pages 35–44, Berkeley, CA.

R. M. Kaplan and J. Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

G. Salton. 1988. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley.

Robust Photo Retrieval Using World Semantics

Hugo Liu*, Henry Lieberman*

* MIT Media Laboratory
Software Agents Group
20 Ames St., E15-320G
Cambridge, MA 02139, USA
{hugo, lieber}@media.mit.edu

Abstract

Photos annotated with textual keywords can be thought of as resembling documents, and querying for photos by keywords is akin to the information retrieval done by search engines. A common approach to making IR more robust involves query expansion using a thesaurus or other lexical resource. The chief limitation is that keyword expansions tend to operate on a word level, and expanded keywords are generally lexically motivated rather than conceptually motivated. In our photo domain, we propose a mechanism for robust retrieval by expanding the *concepts* depicted in the photos, thus going beyond lexical-based expansion. Because photos often depict places, situations and events in everyday life, concepts depicted in photos such as place, event, and activity can be expanded based on our "common sense" notions of how concepts relate to each other in the real world. For example, given the concept "surfer" and our common sense knowledge that surfers can be found at the beach, we might provide the additional concepts: "beach", "waves", "ocean", and "surfboard". This paper presents a mechanism for robust photo retrieval by expanding annotations using a world semantic resource. The resource is automatically constructed from a large-scale freely available corpus of commonsense knowledge. We discuss the challenges of building a semantic resource from a noisy corpus and applying the resource appropriately to the task.

1. Introduction

The task described in this paper is the robust retrieval of annotated photos by a keyword query. By "annotated photos," we mean a photo accompanied by some metadata about the photo, such as keywords and phrases describing people, things, places, and activities depicted in the photo. By "robust retrieval," we mean that photos should be retrievable not just by the explicit keywords in the annotation, but also by other implicit keywords conceptually related to the event depicted in the photo.

In the retrieval sense, annotated photos behave similarly to documents because both contain text, which can be exploited by conventional IR techniques. In fact, the common query enrichment techniques such as thesaurus-based keyword expansion developed for document retrieval may be applied to the photo retrieval domain without modification.

However, keyword expansion using thesauri is limited in its usefulness because keywords expanded by their synonyms can still only retrieve documents directly related to the original keyword. Furthermore, naïve synonym expansion may actually contribute more noise to the query and negate what little benefit keyword expansion may add to the query, namely, if keywords cannot have their word sense disambiguated, then synonyms for all the word senses of a particular word may be used in the expansion, and this has the potential to retrieve many irrelevant documents.

1.1. Relevant Work

Attempting to overcome the limited usefulness of keyword expansion by synonyms, various researchers have tried to use slightly more sophisticated resources for query expansion. These include dictionary-like resources such as lexical semantic relations (Voorhees, 1994), and keyword co-occurrence statistics (Peat and Willet, 1991; Lin, 1998), as well as resources generated dynamically through relevance feedback, like global document analysis (Xu and Croft, 1996), and collaborative concept-based expansion (Klink, 2001).

Although some of these approaches are promising, they share some of the same problems as naïve synonym expansion. Dictionary-like resources such as WordNet (Fellbaum, 1998) and co-occurrence frequencies, although more sophisticated that just synonyms, still operate mostly on the word-level and suggest expansions that are lexically motivated rather than conceptually motivated. In the case of WordNet, lexical items are related through a very limited set of nymic relations. Relevance feedback, though somewhat more successful than dictionary approaches, requires additional iterations of user action and we cannot consider it fully automated retrieval, which makes it an inappropriate candidate for our task.

1.2. Photos vs. Documents

With regard to our domain of photo retrieval, we make a key observation about the difference between photos and documents, and we exploit this difference to make photo retrieval more robust. We make the observation that photos taken by an ordinary person has more structure and is more predictable than the average document on the web, even though that structure may not be immediately evident. The contents of a typical document such as a web page are hard to predict, because there are too many types and genres of web pages and the content does not predictably follow a stereotyped structure. However, with typical photos, such as one found in your photo album, there is more predictable structure. That is, the intended subject of photos often includes people and things in common social situations. Many of these situations depicted, such as weddings, vacations, sporting events, sightseeing, etc. are common to human experience, and therefore have a high level of predictability.

Take for example, a picture annotated with the keyword *"bride"*. Even without looking at the photo, a person may be able to successfully guess who else is in the photo, and what situation is being depicted. Common

sense would lead a person to reason that brides are usually found at weddings, that people found around her may be the groom, the father of the bride, bridesmaids, that weddings may take place in a chapel or church, that there may be a wedding cake, walking down the aisle, and a wedding reception. Of course, common sense cannot be used to predict the structure of specialty photos such as artistic or highly specialized photos; this paper only considers photos in the realm of consumer photography.

1.2.1. A Caveat

Before we proceed, it is important to point out that any semantic resource that attempts to encapsulate common knowledge about the everyday world is going to be somewhat culturally specific. The previous example of brides, churches and weddings illustrates an important point: knowledge that is *obvious* and *common* to one group of people (in this case, middle-class USA) may not be so obvious or common to other groups. With that in mind, we go on to define the properties of this semantic resource.

1.3. World Semantics

Knowledge about the spatial, temporal, and social relations of the everyday world is part of commonsense knowledge. We also call this *world semantics*, referring to the meaning of everyday concepts and how these concepts relate to each other in the world.

The mechanism we propose for robust photo retrieval uses a world semantic resource in order to expand concepts in existing photo annotations with concepts that are, *inter alia*, spatially, temporally, and socially related. More specifically, we automatically constructed our resource from a corpus of English sentences about commonsense by first extracting predicate argument structures, and then compiling those structures into a Concept Node Graph, where the nodes are commonsense concepts, and the weighted edges represent commonsense relations. The graph is structured much like MindNet (Richardson et al., 1998). Performing concept expansion using the graph is modeled as spreading activation (Salton and Buckley, 1988). The relevance of a concept is measured as the semantic proximity between nodes on the graph, and is affected by the strength of the links between nodes.

This paper is structured as follows: First, we discuss the source and nature of the corpus of commonsense knowledge used by our mechanism. Second, a discussion follows regarding how our world semantic resource was automatically constructed from the corpus. Third, we show the spreading activation strategy for robust photo retrieval, and give heuristics for coping with the noise and ambiguity of the knowledge. The paper concludes with a discussion of the larger system to which this mechanism belongs, potential application of this type of resource in other domains, and plans for future work.

2. OMCS: A Corpus of Common Sense

The source of the world semantic knowledge used by our mechanism is the Open Mind Common Sense Knowledge Base (OMCS) (Singh, 2002) - an endeavor at the MIT Media Laboratory that aims to allow a web-community of teachers to collaboratively build a database of "common sense" knowledge.

It is hard to define what actually constitutes common sense, but in general, one can think of it as knowledge about the everyday world that most people within some population consider to be "obvious." As stated earlier, common sense is somewhat culturally specific. Although many thousands of people from around the world collaboratively contribute to Open Mind Common Sense, the majority of the knowledge in the corpus reflects the cultural bias of middle-class USA. In the future, it may make sense to tag knowledge by their cultural specification.

OMCS contains over 400,000 semi-structured English sentences about commonsense, organized into an ontology of commonsense relations such as the following:

- A is a B
- You are likely to find A in/at B
- A is used for B

By semi-structured English, we mean that many of the sentences loosely follow one of 20 or so sentence patterns in the ontology. However, the words and phrases represented by A and B (see above) are not restricted. Some examples of sentences in the knowledge base are:

- Something you find in (a restaurant) is (a waiter)
- The last thing you do when (getting ready for bed) is (turning off the lights)
- While (acting in a play) you might (forget your lines)

The parentheses above denote the part of the sentence pattern that is unrestricted. While English sentence patterns has the advantage of making knowledge easy to gather from ordinary people, there are also problems associated with this. The major limitations of OMCS are four-fold. First, there is ambiguity resulting from the lack of disambiguated word senses, and from the inherent nature of natural languages. Second, many of the sentences are unusable because they may be too complex to fully parse with current parser technology. Third, because there is currently no truth maintenance mechanism or filtering strategy for the knowledge gathered (and such a mechanism is completely nontrivial to build), some of the knowledge may be anomalous, i.e. not common sense, or may plainly contradict other knowledge in the corpus. Fourth, in the acquisition process, there is no mechanism to ensure a broad coverage over many different topics and concepts, so some concepts may be more developed than others.

The Open Mind Commonsense Knowledge Base is often compared with its more famous counterpart, the CYC Knowledge Base (Lenat, 1998). CYC contains over 1,000,000 hand-entered rules that constitute "common sense". Unlike OMCS, CYC represents knowledge using formal logic, and ambiguity is minimized. In fact, it does not share any of the limitations mentioned for OMCS. Of course, the tradeoff is that whereas a community of non-experts contributes to OMCS, CYC needs to be somewhat carefully engineered. Unfortunately, the CYC corpus is not publicly available at this time, whereas OMCS *is* freely available and downloadable via the website (www.openmind.org/commonsense).

Even though OMCS is a more noisy and ambiguous corpus, we find that it is still suitable to our task. By

normalizing the concepts, we can filter out some possibly unusable knowledge (Section 3.2). The impact of ambiguity and noise can be minimized using heuristics (Section 4.1). Even with these precautionary efforts, some anomalous or bad knowledge will still exist, and can lead to seemingly semantically irrelevant concept expansions. In this case, we rely on the fail-soft nature of the application that uses this semantic resource to handle noise gracefully.

3. Constructing a World Semantic Resource

In this section, we describe how a usable subset of the knowledge in OMCS is extracted and structured specifically for the photo retrieval task. First, we apply sentence pattern rules to the raw OMCS corpus and extract crude predicate argument structures, where predicates represent commonsense relations and arguments represent commonsense concepts. Second, concepts are normalized using natural language techniques, and unusable sentences are discarded. Third, the predicate argument structures are read into a Concept Node Graph, where nodes represent concepts, and edges represent predicate relationships. Edges are weighted to indicate the strength of the semantic connectedness between two concept nodes.

3.1. Extracting Predicate Argument Structures

The first step in extracting predicate argument structures is to apply a fixed number of mapping rules to the sentences in OMCS. Each mapping rule captures a different commonsense relation. Commonsense relations, insofar as what interests us for constructing our world semantic resource for photos, fall under the following general categories of knowledge:

1. Classification: A dog is a pet
2. Spatial: San Francisco is part of California
3. Scene: Things often found together are: restaurant, food, waiters, tables, seats
4. Purpose: A vacation is for relaxation; Pets are for companionship
5. Causality: After the wedding ceremony comes the wedding reception.
6. Emotion: A pet makes you feel happy; Rollercoasters make you feel excited and scared.

In our extraction system, mapping rules can be found under all of these categories. To explain mapping rules, we give an example of knowledge from the aforementioned Scene category:

```
somewhere THING1 can be is PLACE1
somewherecanbe
THING1, PLACE1
0.5, 0.1
```

Mapping rules can be thought of as the grammar in a shallow sentence pattern matching parser. The first line in each mapping rule is a sentence pattern. THING1 and PLACE1 are variables that approximately bind to a word or phrase, which is later mapped to a set of canonical commonsense concepts. Line 2 specifies the name of this predicate relation. Line 3 specifies the arguments to the predicate, and corresponds to the variable names in line 1. The pair of numbers on the last line represents the confidence weights given to forward relation (left to right), and backward relation (right to left), respectively, for this predicate relation. This also corresponds to the weights associated with the directed edges between the nodes, THING1 and PLACE1 in the graph representation.

It is important to distinguish the value of the forward relation on a particular rule, as compared to a backward relation. For example, let us consider the commonsense fact, *"somewhere a bride can be is at a wedding."* Given the annotation *"bride,"* it may be very useful to return *"wedding."* However, given the annotation *"wedding,"* it seems to be less useful to return *"bride," "groom," "wedding cake," "priest,"* and all the other things found in a wedding. For our problem domain, we will generally penalize the direction in a relation that returns hyponymic concepts as opposed to hypernymic ones. The weights for the forward and backward directions were manually assigned based on a cursory examination of instances of that relation in the OMCS corpus.

Approximately 20 mapping rules are applied to all the sentences (400,000+) in the OMCS corpus. From this, a crude set of predicate argument relations are extracted. At this time, the text blob bound to each of the arguments needs to be normalized into concepts.

3.2. Normalizing Concepts

Because any arbitrary text blob can bind to a variable in a mapping rule, these blobs need to be normalized into concepts before they can be useful. There are three categories of concepts that can accommodate the vast majority of the parseable commonsense knowledge in OMCS: Noun Phrases (things, places, people), Attributes (adjectives), and Activity Phrases (e.g.: *"walk the dog," "buy groceries."*), which are verb actions that take either no argument, a direct object, or indirect object.

To normalize a text blob into a Noun Phrase, Attribute or Activity Phrase, we tag the text blob with part of speech information, and use these tags filter the blob through a miniature grammar. If the blob does not fit the grammar, it is massaged until it does or it is rejected altogether. Sentences, which contain text blobs that cannot be normalized, are discarded at this point. The final step involves normalizing the verb tenses and the number of the nouns. Only after this is done can our predicate argument structure be added to our repository.

The aforementioned noun phrase, and activity phrase grammar is shown below in a simplified view. Attributes are simply singular adjectives.

```
NOUN PHRASE:
(PREP) (DET|POSS-PRON) NOUN
(PREP) (DET|POSS-PRON) NOUN NOUN
(PREP) NOUN POSS-MARKER (ADJ) NOUN
(PREP) (DET|POSS-PRON) NOUN NOUN NOUN
(PREP) (DET|POSS-PRON) (ADJ) NOUN PREP NOUN

ACTIVITY PHRASE:
(PREP) (ADV) VERB (ADV)
(PREP) (ADV) VERB (ADV) (DET|POSS-PRON) (ADJ) NOUN
(PREP) (ADV) VERB (ADV) (DET|POSS-PRON) (ADJ) NOUN NOUN
(PREP) (ADV) VERB (ADV) PREP (DET|POSS-PRON) (ADJ) NOUN
```

The grammar is used as a filter. If the input to a grammar rule matches any optional tokens, which are in parentheses, then this is still considered a match, but the output will filter out any optional fields. For example, the phrase, *"in your playground"* will match the first rule and the phrase will stripped to just *"playground."*

3.3. Concept Node Graph

To model concept expansion as a spreading activation task, we convert the predicate argument structures gathered previously into a Concept Node Graph by mapping arguments to concept nodes, and predicate relations to edges connecting nodes. Forward and backward edge weights come from the mapping rule associated with each predicate relation. A segment of the graph is shown in Figure 1.

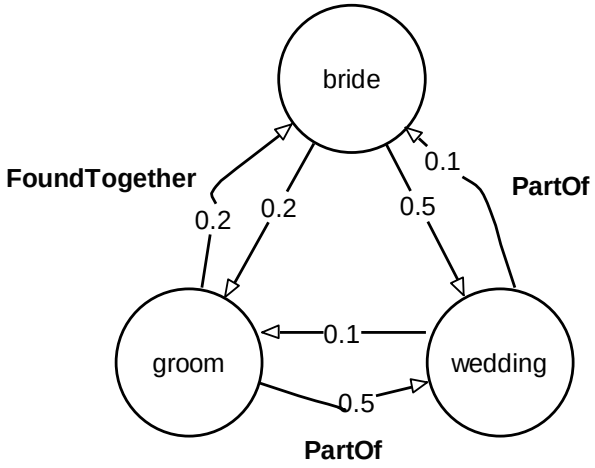


Figure 1. A portion of the Concept Node Graph. Nodes are concepts, and edges correspond to predicate relations.

The following statistics were compiled on the automatically constructed resource:

- 400,000+ sentences in OMCS corpus
- 50,000 predicate argument structures extracted
- 20 predicates in mapping rules
- 30,000 concept nodes
- 160,000 edges
- average branching factor of 5

4. Concept Expansion Using Spreading Activation

In this section, we explain how concept expansion is modeled as spreading activation. We propose two heuristics for re-weighting the graph to improve relevance. Examples of the spreading activation are then given.

In spreading activation, the origin node is the concept we wish to expand (i.e. the annotation) and it is the first node to be activated. Next, nodes one hop away from the origin node are activated, then two levels away, and so on. A node will only be activated if its activation score (AS) meets the activation threshold, which is a tolerance level between 0 (irrelevant) and 1.0 (most relevant). The origin node has a score of 1.0. Given two nodes A and B, where A has 1 edge pointing to B, the activation score of B is given in equation (1).

$$AS(B) = AS(A) * weight(edge(A,B)) \quad \textbf{(1)}$$

When no more nodes are activated, we have found all the concepts that expand the input concept up to our set threshold.

4.1. Heuristics to Improve Relevance

One problem that can arise with spreading activation is that nodes that are activated two or more hops away from the origin node may quickly lose relevance, causing the search to lose focus. One reason for this is noise. Because concept nodes do not make distinctions between different word senses (an aforementioned problem with OMCS), it is possible that a node represents many different word senses. Therefore, activating more than one hop away risks exposure to noise. Although associating weights with the edges provides some measure of relevance, these weights form a homogenous class for all edges of a common predicate (recall that the weights came from mapping rules).

We identify two opportunities to re-weight the graph to improve relevance: reinforcement and popularity. Both of these heuristics are known techniques associated with spreading activation networks (Salton and Buckley, 1988). We motivate their use here with observations about our particular corpus, OMCS.

4.1.1. Reinforcement

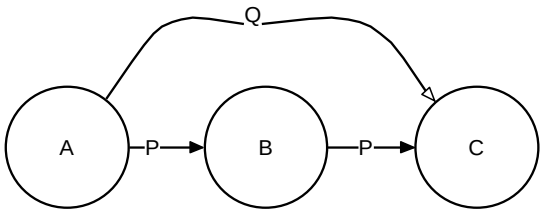


Figure 2. An example of reinforcement

As illustrated in Figure 2, we make the observation that if node C is connected to node A through both paths P and Q, then C would be more relevant to A than had either path P or Q been removed. We call this *reinforcement* and define it as two or more corroborating pieces of evidence, represented by paths, that two nodes are semantically related. The stronger the reinforcement, the higher the potential relevance.

Looking at this in another way, if three or more nodes are mutually connected, they form a cluster. Examples of clusters in our corpus are higher-level concepts like weddings, sporting events, parties, etc., that each have many inter-related concepts associated with them. Within each such cluster, any two nodes have enhanced relevance because the other nodes provide additional paths for reinforcement. Applying this, we re-weight the graph by detecting clusters and increasing the weight on edges within the cluster.

4.1.2. Popularity

The second observation we make is that if an origin node A has a path through node B, and node B has 100

children, then each of node B's children are less likely to be relevant to node A than if node B had had 10 children.

We refer to nodes with a large branching factor as being popular. It happens that popular nodes in our graph tend to either correspond to very common concepts in commonsense, or tend to have many different word senses, or word contexts. This causes its children to have in general, a lower expectation of relevance.

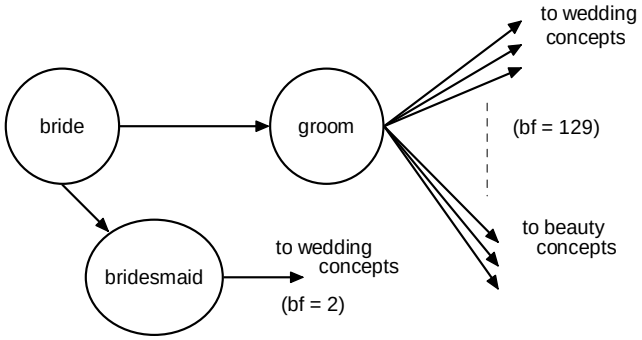


Figure 3. Illustrating the negative effects of popularity

As illustrated in Figure 3, the concept *bride* may lead to *bridesmaid* and *groom*. Whereas *bridesmaid* is a more specific concept, not appearing in many contexts, *groom* is a less specific concept. In fact, different senses and contexts of the word can mean "the groom at a wedding," or "grooming a horse" or "he is well-groomed." This causes *groom* to have a much larger branching factor.

It seems that even though our knowledge is common sense, there is more value associated with more specific concepts than general ones. To apply this principle, we visit each node and discount the weights on each of its edges based on the metric in equation (2). (α and β are constants):

$$newWeight = oldWeight * discount$$
$$discount = \frac{1}{\log(\alpha * branchingFactor + \beta)} \quad (2)$$

4.2. Examples

Below are actual runs of the concept expansion program using an activation threshold of 0.1. They were selected to illustrate what can be commonly expected from the expansions, including limitations posed by the knowledge.

```
>>> expand("bride")
('love', '0.632'), ('wedding', '0.5011')
('groom', '0.19'), ('marry', '0.1732')
('church', '0.1602'), ('marriage', '0.1602')
('flower girl', '0.131') ('happy', '0.131')
('flower', '0.131') ('lake', '0.131')
('cake decoration', '0.131') ('grass', '0.131')
('priest', '0.131') ('tender moment', '0.131')
('veil', '0.131') ('wife', '0.131')
('wedding dress', '0.131') ('sky', '0.131')
('hair', '0.1286') ('wedding bouquet', '0.1286')
('snow covered mountain', '0.1286')

>>> expand('london')
('england', '0.9618') ('ontario', '0.6108')
('europe', '0.4799') ('california', '0.3622')
('united kingdom', '0.2644') ('forest', '0.2644')
('earth', '0.1244')

>>> expand("symphony")
('concert', '0.5') ('music', '0.4')
('theatre', '0.2469')
('conductor', '0.2244')
('concert hall', '0.2244')
('xylophone', '0.1') ('harp', '0.1')
('viola', '0.1') ('cello', '0.1')
('wind instrument', '0.1') ('bassoon', '0.1')
('violin', '0.1')

>>> expand("listen to music")
('relax', '0.4816') ('be entertained', '0.4816')
('have fun', '0.4') ('relaxation', '0.4')
('happy', '0.4') ('hang', '0.4')
('hear music', '0.4') ('dorm room', '0.4')
('understand', '0.4') ('mother', '0.2')
('happy', '0.136')
('get away', '0.136') ('listen', '0.136')
('change psyche', '0.136') ('show', '0.1354')
('dance club', '0.1295') ('frisbee', '0.1295')
('scenery', '0.124') ('garden', '0.124')
('spa', '0.124') ('bean bag chair', '0.124')
```

The expansion of *"bride"* shows the diversity of relations found in the semantic resource. *"Love"* is some emotion that is implicitly linked to brides, weddings, and marriage. Expansions like *"priest", "flower girl,"* and *"groom"* are connected through social relations. *"Wife"* seems to be temporally connected. To *"marry"* indicates the function of a wedding.

However, there are also expansions whose connections are not as obvious, such as *"hair,"* and *"lake."* There are also other expansions that may be anomalies in the OMCS corpus, such as *"tender moment"* and *"snow covered mountain."* These examples point to the need for some type of statistical filtering of the knowledge in the corpus, which is not currently done.

In the last expansion example, the concept of "listen to music" is arguably more abstract than the wedding concept, and so the expansions may seem somewhat arbitrary. This illustrates one of the limitations of any common sense acquisition effort: deciding upon which topics or concepts to cover, how well they are covered, and to what granularity they are covered.

5. Conclusion

In this paper, we presented a mechanism for robust photo retrieval: using a world semantic resource to expand a photo's annotations. The resource was automatically constructed from the publicly available Open Mind Common Sense corpus. Sentence patterns were applied to the corpus, and simple predicate argument structures were extracted. After normalizing arguments into syntactically neat concepts, a weighted concept node graph was constructed. Concept expansion is modeled as spreading activation over the graph. To improve relevance in spreading activation, the graph was re-weighted using heuristics for reinforcement and popularity.

This work has not yet been formally evaluated. Any evaluation will likely take place in the context of the larger system that this mechanism is used in, called (A)nnotation and (R)etrieval (I)ntegration (A)gent (Lieberman et al., 2001) ARIA is an assistive software agent which automatically learns annotations for photos by observing how users place photos in emails and web pages. It also monitors the user as s/he types an email and finds opportunities to suggest relevant photos. The idea of using world semantics to make the retrieval process more robust comes from the observation that concepts depicted in photos are often spatially, temporally, and socially related in a commonsensical way. While the knowledge extracted from OMCS does not give very complete coverage of many different concepts, we believe that what concept expansions *are* done have added to the robustness of the retrieval process. Sometimes the concept expansions are irrelevant, but because ARIA engages in opportunistic retrieval that does not obstruct the user's task of writing the email, the user does not suffer as a result. We sometimes refer to ARIA as being "fail-soft" because good photo suggestions can help the task, but the user can ignore bad photo suggestions.

Robust photo retrieval is not the only IR task in which semantic resources extracted from OMCS have been successfully applied. (Liu et al., 2002) used OMCS to perform inference to generate effective search queries by analyzing the user's search goals. (Liu and Singh, 2002) uses the subset of causal knowledge in OMCS to generate crude story scripts.

In general, the granularity of the knowledge in OMCS can benefit any program that deals with higher-level social concepts of the everyday world. However, because of limitations associated with this corpus such as noise, ambiguity, and coverage, OMCS is likely to be only useful at a very shallow level, such as providing an associative mechanism between everyday concepts or performing first-order inference.

Future work is planned to improve the performance of the mechanism presented in this paper. One major limitation that we have encountered is noise, stemming from ambiguous word senses and contexts. To overcome this, we hope to apply known word sense disambiguation techniques to the concepts and the query, using word sense co-occurrence statistics, WordNet, or LDOCE. A similar approach could be taken to disambiguate meaning contexts, but it is less clear how to proceed.

Another point of future work is the migration from the sentence pattern parser to a broad coverage parser so that we can extract more kinds of commonsense relations from the corpus, and make more sentences "usable."

6. Acknowledgements

We thank our colleagues Push Singh, and Kim Waters at the MIT Media Lab, Tim Chklovski at the MIT AI Lab, and Erik Mueller at IBM, who are also working on the problem of commonsense, for their contributions to our collective understanding of the issues. We would especially like to thank Push for directing OMCS and for his advocacy of commonsense.

7. References

Fellbaum, C. (ed.), 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Klink, S., 2001. Query reformulation with collaborative concept-based expansion. *Proceedings of the First International Workshop on Web Document Analysis*, Seattle, WA.

Lenat, D., 1998. *The dimensions of context-space*, Cycorp technical report, www.cyc.com.

Lieberman, H., Rosenzweig E., Singh, P., 2001. Aria: An Agent For Annotating And Retrieving Images, *IEEE Computer*, July 2001, pp. 57-61.

Lin, D., 1998. Using collocation statistics in information extraction. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, San Francisco, CA, http://www.muc.saic.com.

Liu, H., Lieberman, H., Selker, T., 2002. GOOSE: A Goal-Oriented Search Engine With Commonsense. *Proceedings of the 2002 International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain.

Liu, H., Singh, P., 2002. MAKEBELIEVE: Using Commonsense to Generate Stories. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-02)* -- Student Abstract. Seattle, WA.

Open Mind Common Sense Website and Corpus. Available at: www.openmind.org/commonsense.

Peat, H.J. and Willett, P., 1991. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the ASIS*, 42(5), 378-383.

Richardson, S.D., Dolan, W.B., Vanderwende, L., 1998. MindNet: Acquiring and Structuring Semantic Information from Text. *Proceedings of the joint ACL and COLING conference*, 1098-1102, Montreal.

Salton, G., and Buckley, C., 1988. On the Use of Spreading Activation Methods in Automatic Information Retrieval. *Proceedings of the 11th Ann. Int. ACM SIGIR Conf. on R&D in Information Retrieval (ACM)*, 147-160.

Singh, P., 2002. The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.* Palo Alto, CA, AAAI.

Voorhees, E., 1994. Query expansion using lexical-semantic relations. *Proceedings of ACM SIGIR Intl. Conf. on Research and Development in Information Retrieval*, pages 61—69.

Xu, J., and Croft, W.B., 1996. Query Expansion Using Local and Global Document Analysis. Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4—11.

Managing Synonymy and Polysemy in a Document Retrieval System Using WordNet

Claude de Loupy[*]
Marc El-Bèze[**]

[*] Sinequa
51-54, rue Ledru-Rollin
94200 Ivry-sur-Seine
France
loupy@sinequa.com
http://www.sinequa.com

[**] Laboratoire Informatique d'Avignon
B.P. 1228 Agroparc
339 Chemin des Meinajaries
84911 Avignon Cedex 9
France
marc.elbeze@lia.univ-avignon.fr
http://www.lia.univ-avignon.fr/

Abstract

This paper reports several experiments of document retrieval with TREC-6 using semantic knowledge. In a first set of experiments, synonyms and hyponyms given by WordNet are used in order to enrich queries. A small improvement is shown. The second set uses a word sense disambiguation system in order to cope with polysemy. There is almost no modification of performances but this is an important result considering Sanderson's results. Our system performs at 72% of accuracy when Sanderson concludes a system performing at less than 90% degrades results. When using both query enrichment and WSD, the improvements are a little better, especially for the first document retrieved. Lastly, a small set of experiments using specialized thesauri is presented, showing important improvements.

Keywords

Document Retrieval, Word Sense Disambiguation, Synonymy, Polysemy, WordNet, HMM

1 Introduction

From the beginning of automatic Document Retrieval (DR), researchers have tried to use thesaurus. But results were often disappointing: Salton (1968) used the Harris Synonym Thesaurus and noted a fall of performances. both Harman (1988) and Voorhees (1993, 1994) using WordNet, came to the same conclusion, even if Harman noted that when the user is involved in the process, results are improved.

In this paper, we report several experiments using TREC-6 (Harman, 1997) for evaluation, WordNet (Miller *et al.*, 1990) as a semantic lexicon and a Word Sense Disambiguation (WSD) system trained on SemCor (Miller *et al.*, 1993). The results of these experiments contradict some widespread ideas and some conclusions of other experiments.

The DR system used is described in section 2. In section 3 several experiments using query enrichment with synonyms or hyponyms from WordNet are analyzed. In section 4, the impact of WSD in DR is shown. Section 5 reports experiments using both information and section 6 reports the use of specialized thesauri.

2 The Document Retrieval system used

The DR system used for these experiments is *IndeXal* (Loupy *et al.*, 1998a). The similarity measure is the one proposed by Harman (1986) with a slight modification:

$$(1) \quad score(d) = \frac{\sum_{x \in d \cap q} TF_d(x) \cdot IDF(x)}{\sum_{x \in q} TF_d(x) \cdot IDF(x)}$$

where $score(d)$ is the score of document d according to the query, and:

$$(2) \quad IDF(x) = -\log\left(\frac{n(x)}{N}\right)$$

$$(3) \quad TF_d(x) = K + (1-K) \cdot \frac{\log(O_d(x))}{\log(L_d)}$$

with: $n(x)$ the number of documents containing x, N the total number of documents, $O_d(x)$ the number of occurrences of x in d, L_d the length of document d, and K a coefficient (here is the modification). This coefficient is used to determine the relative importance of *IDF* and *TF*. Best scores are obtained with $K = 0.3$. In this paper, the results are evaluated on TREC-6 (Voorhees & Harman, 1997). Only *Titles* were used (that is 1 to 4 words queries). The results of table 1 will serve as reference for comparison with the other results. *stem* represents the results obtained with a classic stemming procedure[1] and *lem* the ones obtained with a POS tagging system called

[1] We used Porter's stemmer (Porter, 1980)

ECSta (Spriet & El-Bèze, 1997) and a lemmatization. The performances for French are good (96.5% of efficiency). We trained the tagger on the SemCor which is a very small corpus. The final performances are only 88.8% of correct assignation. This seems very weak, but considering only the tagging of content words (nouns, verbs, adjectives and adverbs), the error rate is only 3.9%. This seems sufficient for the following experiments.

	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6

Table 1: Basic results

Here, we chose to evaluate the different strategies using only the following statistics given in TREC:

- the number of relevant document retrieved (*Rel-Ret*)
- the average precision (*Av-Prec*)
- the precision at 5, 10, 20 and 100 documents retrieved
- the R-precision (*R-Prec*) that is the precision when there are as many documents returned by the system than relevant ones.

Though stemming seems to be the most efficient strategy, we can see that the precision of the first retrieved documents increases with lemmatization. We think that the precision of the first retrieved documents is the most important for evaluation because they are the documents a user will read and they can be used in an automatic relevance feedback procedure. So, lemmatization does not seem to be a bad strategy. But it would be interesting to improve performances concerning the other statistics, particularly for the precision when 20 documents are retrieved.
In the following experiments, enrichment and disambiguation procedures are used after lemmatization.

3 Using WordNet to Enrich Queries

Smeaton *et al.* (1995) showed relevant documents do not necessarily contain words of the query. One way to improve DR systems performances is to enrich the queries with synonyms or hyponyms.

3.1 Why are synonyms important?

Figure 1 shows the sets of documents containing "*woman*" or "*parliament*" or both terms or none of them and their intersection with the set of relevant documents for query *321* ("*woman in parliament*"). We can see that 10% of relevant documents do not contain the terms "*woman*" and "*parliament*".
It is legitimate to expect that the query enrichment should help the DR systems to retrieve these 10%. Using query enrichment with synonyms and hyponyms, Smeaton *et al.* (1995) retrieved 5% of relevant documents of TREC-3 (Harman, 1994) that do not contain any word of the queries. The following sections show experiments on TREC-6 using query enrichment with synonyms and hyponyms from WordNet 1.5 (Miller *et al.*, 1993).

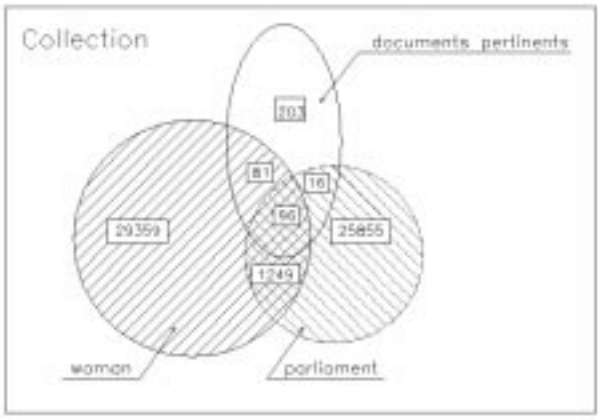


Figure 1: Distribution of documents for request 321

3.2 Presentation of the method

3.2.1 Similarity with enrichment

Enrichment is made at the word level. If a word x of the query has 2 synonyms (y and z), x is replaced by $X = (x, \alpha \cdot y, \alpha \cdot z)$ where $\alpha \in [0,1]$ indicates the importance given to the synonyms compared with the original word. So, we create a pseudo-word X with

(4) $$n(X) = \sum_d C(x,d)$$

with $C(x,d)=1$ if the document d contains x and $C(x,d)=\alpha$ if d does not contain x but contains y or z.
and

(5) $$O_d(X) = O_d(x) + \alpha \cdot O_d(y) + \alpha \cdot O_d(z)$$

It is very important to note that synonyms is taken into account for the calculation of *IDF* and *TF*. Usually, in query enrichment systems, words are added to the query as if they were independent. So each added word has its own *IDF* and *TF*.

3.3 Using Synonyms

In order to enrich queries, we used WordNet 1.5 synsets. 91 591 synsets are given in WordNet 1.5.

3.3.1 A single sense

In this first experiment, only monosemic words are expanded and the expansion concerns only monosemic synonyms. Therefore, polysemy has no influence on the results. The following table gives the results obtained according to the weight α ($\alpha = 0$ corresponds to the lem-basic results and $\alpha = 1$ means that synonyms are as important as original words).
Firstly, we can observe that modifications of the results are very small. But this is a very important observation. It is usually said that the use of synonyms decrease precision and here we can see that it is not the case.
Actually, only 22 queries are concerned by this enrichment. Compared with lemmatization, the performances are increased for 10 of them and decreased for the others (if we consider the average precision). If we

take $\alpha = 0.5$, the average precision is slightly increased (0.3) compared with *Av-prec* but this is not significant. The important fact is that all the decreases in average precision are lower that 1% (absolute values) when the query *317* ("*Unsolicited Faxes*") shows a 7.1 gain if it is enriched by "*unsought*" and "*facsimile*". The other increases are smaller than 4%.

α	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
0.1	1985	21.2	39.6	36.2	30.5	16.8	25.7
0.3	1988	21.3	39.6	36.2	30.5	16.8	25.8
0.5	1988	21.3	40.4	36.6	30.6	16.8	25.8
0.7	1986	21.4	40.4	36.8	30.6	16.8	25.8
0.9	1984	21.3	40.0	36.6	30.9	16.8	25.5
1	1981	21.0	39.2	36.2	30.9	16.8	25.5

Table 2: Enrichment of monosemic words with monosemic synonyms

Concerning query *302*, it is important to note that the gain (+2.8%) is not strictly due to synonymy enrichment. The query is: "*poliomyelitis post polio*". The terms *polio* and *poliomyelitis* are synonyms in WordNet. So, after enrichment, the query is: "(*poliomyelitis* OR *polio*) *post* (*polio* OR *poliomyelitis*)". There is no addition of words, but the calculation of scores is modified. This suggests the system should benefit of a modification of the similarity measure presented in section 2 (formulae 1, 2 and 3).

3.3.2 Several senses

In this section, we want to take into account the number of senses of original words and synonyms in order to see if it is interesting to enrich polysemic words. Table 3 gives the results of an enrichment according to the maximum of senses (*n*) an enriched word and its synonyms have.

n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
1	1988	21.3	40.4	36.6	30.6	16.8	25.8
2	1996	21.4	40.8	36.6	30.7	17.0	26.0
3	1991	21.3	41.2	36.2	30.5	16.9	25.9
4	1967	21.3	40.8	36.0	30.1	16.8	25.7
5	1961	21.4	40.8	36.4	30.3	16.7	26.0
6	1964	21.4	40.8	36.4	30.5	16.7	26.0
7	1957	21.2	40.4	35.6	30.4	16.6	25.8
8	1960	21.2	40.4	35.2	30.5	16.5	25.7
9	1959	21.1	40.0	34.6	30.2	16.3	25.6
∞	1959	21.1	40.0	34.4	30.2	16.3	25.6

Table 3: Enrichment of polysemic words with polysemic synonyms

Here again, there are almost no differences between the basic lemmatization results and the one obtained after enrichment. Nevertheless, it is important to note that there is no decrease of performances when the words which have 3 or less senses are enriched with words which have 3 or less senses. This result will be used in the section 4.

3.4 Using Hyponyms

Another way to enrich queries is to use hyponyms instead of synonyms. The following table gives the results of such an enrichment according to the maximum number of senses (*n*) a word must have to be enriched by its hyponyms (if they also have less than *n* senses).

n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
1	1999	21.4	39.6	36.2	30.9	16.8	25.7
2	2015	21.3	40.4	36.4	30.9	17.1	25.8
3	2026	21.4	41.6	36.2	31.5	17.0	25.9
4	2004	21.3	41.2	36.2	31.2	16.8	25.7
5	2003	21.3	41.2	36.4	31.7	16.8	25.9
syn	1991	21.3	41.2	36.2	30.5	16.9	25.9

Table 4: Enrichment of polysemic words with polysemic hyponyms

The differences are more important here. 32 queries are modified by this enrichment. For 20 queries, performances are increased (up to 9.2 % in absolute value) and for 12 of them performances decrease (down to –6.5 %). In fact, performances are better when using hyponyms instead of synonyms.
It is important to note that the performances for the first 20 documents are approximately the same using hyponyms or stemming.

4 WSD and DR

Polysemy is a very important problem in DR. In this section, we start by a reminder of some important previous experiments. Then, we shall present our own experiments.

4.1 Important previous works[2]

The most cited work concerning the use of WSD for DR is (Sanderson, 1994). Sanderson's conclusion is that a WSD system performing with less than 90% of accuracy decreases results of DR. This is really a problem because the two Senseval evaluations (Kilgarriff & Palmer, 2000) show that the performances of such systems is less than 80%.
This work has been criticized by Schütze and Pedersen (1995) because the use of pseudo-words (Yarowsky, 1993) by Sanderson does not fit the real behavior of polysemic words. They even showed an improvement of performances using WSD on TREC evaluation. But their system is based on automatic construction of a thesaurus.
Gonzalo *et al.* (1998b) used the SemCor (Miller *et al.*, 1993) in order to build an evaluation framework where the importance of WSD and synonymy can be easily evaluated. They report a great improvement of performances. This is encouraging but not really a proof. The evaluation corpus is very special: queries were built manually as abstracts of the SemCor documents and they

[2] A more precise description of previous works can be found in Sanderson (2000).

consider there is only one relevant document for a "query".
They also evaluated the influence of disambiguation errors, confirming the results of Sanderson: 10% of wrong disambiguation leads to a decrease in DR results. But, using both WSD and synonymy enrichment, the tolerance of errors is very much higher: with a WSD system performing at 70%, performances are increased and even with 40% of good identification, performances are stable. These results are a bit strange but quiet encouraging for further experiments.
In a further paper, Gonzalo *et al.* (1999) reproduced the Sanderson's experiments using pseudo-words and found a threshold of 75% instead of the 90% expected. This result is more in agreement with the ones of this paper.
The next sections present the use of a complete WSD system in a TREC experiment. We show that, even if performances are not increased, a quite basic system performing between 71.5% and 74.6% of accuracy does not degrade results.

4.2 Presentation of the Method

In section 3.2.2, we saw that enriching original words with synonyms even when they have three senses could be interesting. In this section, we use a WSD system in order to choose the most probable one, two or three senses for words according to their contexts.
The WSD system (Loupy *et al.*, 1998b) is based on HMM. A Baum-Welch algorithm (Baum *et al.*, 1970) is used in order to keep several senses. This is important for document retrieval in view of the following facts:

- the WSD system can do mistakes (see performances below)
- even if the sense of a word is obvious, the other senses are often kept in mind
- since WordNet senses are very fine grained (41 senses for the verb "*run*"), keeping several senses can be useful in order to represent a coarser sense which do not exist.
- it is sometimes impossible to disambiguate a polysemic word (Kilgarriff, 1994) even for a human being.

The HMM model were trained on the SemCor and its performances were evaluated using 95% for training and 5% for tests. The following table gives the scores when 1, 2 or 3 senses are kept, considering all words (*all*) or only ambiguous ones (*amb*). Moreover, two results are given for each case. The first one corresponds to an evaluation when part-of-speech is known (real evaluation of WSD) and the second one when this POS is not known (real world). The model is a bigram one. With unisem, the performances are slightly inferior (of about 0.4).

		1 sense		2 senses		3 senses	
		all	*amb*	*all*	*amb*	*all*	*amb*
POS	**known**	74.6%	62.5%	87.7%	78.0%	92.9%	83.2%
	unk.	71.5%	59.7%	84.5%	74.9%	89.8%	79.6%

Table 5: Performances of the WSD system

So, the performances are really lower than the one given by Sanderson when he said that a WSD system must perform at 90% or more.

4.3 A simple use of disambiguation

If we keep 3 senses during disambiguation, there are many ways to use it. Figure 2 shows the combinations between a disambiguated query and a disambiguated document.

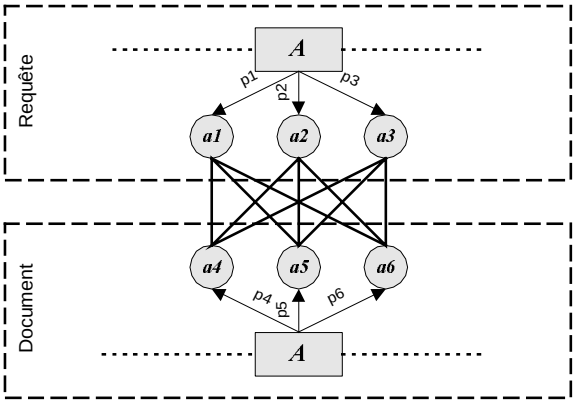


Figure 2 : Combinations between query and document when using disambiguation

Several combinations were tested. Table 6 gives the results. The first line (*all*) gives the results when no disambiguation is made. The other lines (*m-n*) represent a disambiguation where *m* senses are kept in the query and *n* senses are kept in the documents.

m-n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
1-1	1976	20.9	41.2	36.0	30.3	16.8	25.1
1-2	1978	21.1	41.2	35.6	30.6	16.7	25.4
1-3	1979	21.1	40.4	35.8	30.6	16.8	25.5
2-1	1979	21.1	40.8	36.2	30.5	16.8	25.4
2-2	1974	21.2	41.2	36.2	31.2	16.8	25.6
2-3	1969	21.2	40.8	36.2	31.2	16.7	25.6
3-1	1983	21.1	40.4	36.0	30.7	16.8	25.6
3-2	1969	21.2	40.8	36.6	31.4	16.8	25.6
3-3	1971	21.2	40.8	36.4	31.4	16.7	25.6

Table 6: Results of a simple use of WSD in a DR system

Performances are almost the same with WSD (whatever the strategy is) and without. But, here again, there is a very tiny improvement for the first documents retrieved.
If we consider only Average Precision for the *1-1* strategy, results are improved for 24 queries and decreased for only 10 queries. But, while no query is improved by more than 1%, the query *339* ("*Alzheimer's drug treatment*") decreases by 9.5%. The fall of precision of the other queries is less than 1.3%.
So, even if we consider that the problem of the query *339* is an "accident", improvements are very poor. But, we can also conclude that a WSD system performing at 72% does not decrease results of a DR system contrary to what Sanderson claims.

Another interesting point is that there is almost no modification of recall.

4.4 Using sense probability from WSD system

Previous experiments were made without taking into account the probabilistic information (probability of each of the three senses) given by the WSD system. It should be interesting to use them. The similarity measure is the same as the one given in section 2 but the way the number of occurrences is counted is modified:

(6) $$n(x)=S_q(x)\cdot\sum_d Max\left(S_d(x,i)\right)$$

where $S_d(x,i)$ is the probability given by the WSD system to the word-sense x at the position i and $S_Q(x)$ the probability of the word-sense x in the query.

(7) $$O_d(x)=S_Q(x)\cdot\sum_i S_d(x,i)$$

Table 7 gives the results of such a heuristic.

	m-n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
	stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
	all	1984	21.1	39.6	36.0	30.7	16.8	25.6
sense	**1-1**	1976	20.9	41.2	36.0	30.3	16.8	25.1
	2-2	1974	21.2	41.2	36.2	31.2	16.8	25.6
	3-3	1971	21.2	40.8	36.4	31.4	16.7	25.6
req	**1-1**	1913	20.0	39.2	33.6	29.4	16.2	23.6
	2-2	1921	20.3	39.6	34.8	29.9	16.2	24.0
	3-3	1929	20.3	39.6	35.0	29.8	16.2	24.0
doc	**1-1**	1859	18.6	37.2	31.6	27.5	15.1	22.3
	2-2	1881	18.7	36.0	31.8	28.5	15.2	23.2
	3-3	1884	18.8	37.2	32.4	28.5	15.1	23.3
req+doc	**1-1**	1774	17.8	36.4	31.4	26.4	14.4	21.1
	2-2	1777	17.9	36.4	31.8	26.5	14.5	21.2
	3-3	1780	17.9	36.4	31.6	26.6	14.5	21.2

Table 7: Results of using WSD in a DR system taking probabilities into account

The lines *sens* give the results reported in section 4.3 (probabilities are not involved in scores). The lines *req* report the use of WSD probabilities for queries only, *doc* for documents only and *req+doc* for both queries and documents.

We can see that the results have decreased. This is very surprising. Another heuristic may help us to overcome this problem.

5 Using Both WSD and Query Enrichment

In the previous sections, we use query enrichment and WSD in separate experiments. In this section, we shall combine both strategies. The following tables show the performances obtained when one, two or three senses are kept after WSD.

m-n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
syn 1-1	1999	21.4	39.6	36.2	30.9	16.8	25.7
wsd 1-1	1976	20.9	41.2	36.0	30.3	16.8	25.1
syn+wsd	1971	21.1	42.4	36.2	30.2	16.8	25.6

Table 8: combining enrichment and WSD with one sense

m-n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
syn 2-2	2015	21.3	40.4	36.4	30.9	17.1	25.8
wsd 2-2	1974	21.2	41.2	36.2	31.2	16.8	25.6
syn+wsd	1968	21.4	42.4	36.8	31.4	16.8	26.0

Table 9: combining enrichment and WSD with two senses

m-n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
syn 3-3	2026	21.4	41.6	36.2	31.5	17.0	25.9
wsd 3-3	1971	21.2	40.8	36.4	31.4	16.7	25.6
syn+wsd	1968	21.3	42.0	36.8	31.4	16.8	25.7

Table 10: combining enrichment and WSD with three senses

The results show little improvements when keeping 2 or 3 senses and enriching with WordNet synonyms. Of course, the question is: is the gain interesting compared to the cost?

6 Combining synonyms and stemming

As the use of synonyms does not show any improvement, another possibility is to use both information. The following table gives the results of this strategy.

n	Rel-Ret	Av-Prec	5	10	20	100	R-Prec
stem	2068	22.3	38.2	34.8	31.9	17.2	26.1
lem	1984	21.1	39.6	36.0	30.7	16.8	25.6
syn 3-3	2026	21.4	41.6	36.2	31.5	17.0	25.9
lem-stem	2124	23.1	39.2	36.0	31.7	17.9	26.9
lem-stem-syn	2140	22.7	40.8	36.0	32.8	17.3	26.1

Table 11: Results of using both stemming and synonymy enrichment

In this table, we can see that the use of both lemmatization and stemming is more interesting than using one of these strategy alone. The strategy using all these information (*lem-stem-syn*) gives better results than *stem* and *lem* for all statistics. It seems to be an interesting strategy although the precision for 5 documents is lower than the use of synonyms.

Other experiments should be done to evaluate the performances of hyponyms used with stems and synonyms.

7 Using expert knowledge

It is clear for all that the use of expert knowledge should improve performances of DR systems (Anand *et al.*, 1995). For this experiment, a specialized lexicon has been built for each of the ten first queries of TREC-6. The time necessary for this construction is more or less 5 minutes per thesaurus. The built lexicons are, therefore, very small. It is clear that, we never looked at relevant documents to search for relevant terms. Words linked to the words of a query were added to this query. The following list gives the words used for each of the ten queries:

301: international organized (crime drug prostitution cocaine ectasy extasy heroin trafficking traffic terrorism terrosrist criminal mafia maffia triad tong cartel)

302: (poliomyelitis polio brunhilde lansing léon paralysis) post (polio poliomyelitis brunhilde lansing léon paralysis)

303: hubble (telescope space_telescope infrared_telescope optical_mirror space black_hole invisible_space big_bang) (achievement accomplishment)

304: endangered (specie coinage mintage) (mammal panda whale)

305: most (dangerous unsafe grave graver gravest grievous) (vehicle car bus highway road)

306: (african africa angola angolan luanda namibia namibian windhoek bostwana gaborome swaziland mbabame lesotho maseroni south_africa cape_town zimbabwe zimbabwean harare zambia zambian luzaka tanzania tanzanian dar_es_salamm burundi burundian bujumbura uganda ugandan kamdala rwanda rwandan kinshasa congo congolese brazzaville gabon gabonese libreville cameroon cameroonian yaoundé nigeria nigerian abuja chad chadian djamena ndjamena sudani sudanese khartoum ethiopia ethiopian addis_abeba eritrea eritrean asimara somalia somalian mogadishu egypt egyptian cairo libya libyan tripoli tunisia tunisian tunis algeria algerian algiers morocco moroccan rabat mauritania mauritanian nouskshott senegal senegalese dakar mali bamako sierra_leone freetown madagascar madagascana madagascan antananarivo) civilian (death kill war killed killing)

307: (new newer newest) hydroelectric (project undertaking task task projection)

308: (implant implantation) (dentistry dentist tooth)

309: rap music ((crime drug prostitution cocaine ectasy extasy heroin trafficking traffic terrorism terrosrist criminal mafia maffia triad tong cartel)

310: (radio phone) wave brain cancer

We can see, for example, *dentistry* is associated with *dentist* and *tooth* and *vehicle* with *car*, *bus*, *highway* and *road*.

Table 11 gives the results obtained.

4 values are studied: number of relevant documents retrieved (*Rel-Ret*), precision for 20 document retrieved (*20*), average precision and R-precision. They are compared in 3 experiments: the basic one (*bas* - see section 2), query enrichment by WordNet synonyms (*syn* - see section 3.3.2) and query enrichment using expert knowledge (*use* - synonyms, hyponyms, see also links). The last figure represent the gain using specialized thesaurus (*use-bas*).

We can see that, in almost all cases, a specialized thesaurus increases performances. For query *306*, the gain is only due to a very simple geographic thesaurus.

		301	**302**	**303**	**304**	**305**	**306**	**307**	**308**	**309**	**310**
Rel-Ret	lem	88	64	10	97	5	124	155	3	1	6
	syn	88	64	10	97	5	123	151	3	1	6
	use	108	65	10	103	3	165	150	4	1	6
	use-lem	+20	+1	=	+6	-2	+41	-5	+1	=	=
20	lem	45.0	75.0	10.0	35.0	0.0	35.0	50.0	15.0	0.0	10.0
	syn	45.0	70.0	15.0	35.0	0.0	65.0	50.0	15.0	0.0	10.0
	use	55.5	75.0	10.0	40.0	5.0	75.0	45.0	20.0	0.0	10.0
	use-lem	+10	=	=	+5	+5	+10	-5	+5	=	=
Av. Prec.	lem	5.8	62.9	19.6	10.8	0.2	13.4	26.2	58.3	0.2	7.9
	syn	5.7	65.2	19.5	11.0	0.3	13.3	25.4	58.3	0.2	7.9
	user	9.5	65.6	22.2	16.0	0.4	24.7	24.9	75.4	0.4	7.9
	use-lem	+3.7	+2.7	+2.6	+5.2	+0.2	+11.3	-1.3	+17.1	+0.2	=
R-Prec.	lem	15.2	60.0	10.0	26.5	0.0	21.7	38.1	50.0	0.0	15.4
	syn	15.4	63.1	10.0	26.5	0.0	22.9	37.1	50.0	0.0	15.4
	use	19.9	63.1	10.0	31.6	5.7	41.0	37.1	75.0	0.0	15.4
	use-lem	+4.7	+3.1	=	+5.1	+5.7	+19.3	-1.0	+25.0	=	=

Table 11: Using expert knowledge for TREC queries

8 Conclusion

The experiments reported in this paper were only made on TREC-6. In order to confirm the results, they should be applied on other evaluation frameworks. Moreover, it would be interesting to use different heuristics, specially in section 4.4. But these results already lead to several conclusions:

- Using synonymy enrichment not necessarily decreases precision.
- Using WSD not necessarily decreases recall.
- A WSD system performing at 72% of accuracy does not necessarily degrades results, contrary to Sanderson's conclusions.
- The contribution of synonymy enrichment and WSD can be very poor compared to the amount of work necessary to build the necessary resources and tools.
- The combination of resources gives the best results.
- The use of specialized resources can be very useful in order to improve performances.

Of course, it seems that the "cost" is too important regarding the small improvement. In fact, the problem may come from the knowledge source, that is WordNet. It has been often criticized for DR applications for the following reasons:

- Semantic links are only possible in the same part of speech (for instance, there is no link between "*to cook*" and "*cooking*") (Gonzalo *et al.*, 1998a).
- There is no link between words of the same domain. Fellbaum *et al.* (1996) point out that the words *tennis*, *racket*, *ball* and *tennis player* have no relation.

- Senses are too fine grained (Palmer, 1998).

Another problem is that some senses are ignored. Shütze and Pedersen (1995) noticed the sense *horse race* is ignored for the word *derby* which is only tagged as a *hat*. According to them, this is an argument to use specialized automatically built resources instead of a general manually built one. An alternative solution should be find at the intersection of the two worlds: using lexical resources to have a basic knowledge and learn some relations from corpus while indexing.
One very important fact is that it is almost every time beneficial to involve users in the whole process. The next step of information retrieval will be to interact with the user. And one of the most interesting way to do that is to use lexical resources (automatically built or not) and systems performing WSD in order to help the user and to save him time. Particularly, it should be interesting to manually disambiguate queries.

9 References

Anand S. S., Bell D. A., Hughes J. G. (1995). The role of domain knowledge in data mining. in *Proceedings of International Conference on Information and Knowledge Management* (CIKM'95), pp. 37-43.

Baum L.E., Petrie T., Soules G., Weiss N. (1970). *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*. AMS Vol. 41, No. 1, pp. 167-171.

Fellbaum C., Grabowski J., Landes S. , Baumann A. (1996). Matching words to senses in WordNet: naive vs expert differentiation of senses. *WordNet: An electronic lexical database and some of its applications*, (editor C. Fellbaum), MIT Press, Cambridge, USA.

Gonzalo J., Verdejo F., Peters C. , Calzolari N. (1998a). Applying EuroWordNet to Cross-Language Text Retrieval. *Computers and the humanities*, Special Issue on EuroWordNet.

Gonzalo J., Verdejo F., Chugur I., Cigarran J. (1998b). Indexing with WordNet synsets can improve text retrieval. in *Proceedings of the Workshop on Usage of WordNet for Natural Language Processing*.

Gonzalo, J., A. Peñas and F. Verdejo, 1999, Lexical Ambiguity and Information Retrieval Revisited. in *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, pp 195-202.

Harman D. (1986). An experimental study of factors important in document ranking. in *Proceedings of ACM Conference on Research and Development in Information Retrieval*. Pisa, Italy.

Harman D. (1998). Towards interactive query expansion. in *Proceedings of the 11th Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, pp. 321-331. Grenoble, France.

Harman D. (1994). Overview of the Third Text REtrieval Conference (TREC-3). *NIST Special Publication 500-226*, p 1.

Kilgarriff A., Palmer M. (Editors) (2000). Special Issue on SENSEVAL. *Computers and the Humanities*.

Loupy C. de, Bellot P., El-Bèze M. , Marteau P.F. (1998a). *Query expansion and classification of retrieved documents*. Seventh Text Retrieval Conference (TREC-7), pp. 443-450. Gaithersburg, Maryland, USA.

Loupy C. de, El-Bèze M., Marteau P.-F. (1998b). *Word Sense Disambiguation using HMM Tagger*. First International Conference on Language Resources & Evaluation, pp. 1255-1258. Granada, Spain.

Miller G.A., Beckwith R., Fellbaum C., Gross D., Miller K. (1990) *Introduction to WordNet: An online lexical database*. International Journal of Lexicography 3 (4), pp. 235-244.

Miller G. A., Leacock C., Randee T., Bunker R. (1993) *A semantic concordance*. 3rd DARPA Workshop on Human Language Technology, pp. 303-308. Plainsboro, New Jersey, USA.

Palmer M. (1998). Are WordNet sense distinctions appropriate for computational lexicons? in *Proceedings of SENSEVAL Workshop*. Herstmonceux Castle, England.

Porter M.F. (1980). An algorithm for suffix stripping. in *Program* 14 (3), pp. 130-137.

Salton G. (1968). *Automatic information organization and retrieval*. McGraw-Hill Book Company.

Sanderson M. (1994). Word sense disambiguation and information retrieval in *Proceedings of the 17th annual international ACM-SIGIR conference on Research and development in information retrieval*, pp. 142-151.

Sanderson (2000). Retrieving with good sense. in *Information Retrieval* Vol. 2 No. 1, pp. 49-69.

Schütze H., Pedersen J. (1995). Information retrieval based on word senses. in *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 161-175.

Smeaton A. F., Kelledy F., O'Donnel R. (1995). *TREC-4 experiments at Dublin City University: Thresholding posting lists, query expansion with WordNet and POS tagging of Spanish*. TREC-4, pp. 373-390.

Spriet T., El-Bèze M. (1997). *Introduction of rules into a stochastic approach for language modeling*. Computational Models for Speech Pattern Processing, NATO ASI Series F, editor K.M. Ponting.

Voorhees E. M. (1993). Using WordNet to disambiguate word senses for text retrieval. in *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 171-180. Pittsburg, USA.

Voorhees E. M. (1994). Query expansion using lexical-semantic relations. in *Proceedings of the 17th annual international ACM-SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'94), pp. 61-69.

Voorhees E., Harman D. (1997). *Overview of the sixth Text Retrieval Conference*. Sixth Text REtrieval Conference, NIST Special Publication 500-240, pp. 1-24. Gaithersburg, MD, USA.

Yarowsky D. (1993). One sense per collocation. in *Proceedings of the ARPA Human Language Technology Workshop*, pp. 266-271, San Francisco, USA.

The Semantic Wildcard

Rada F. MIHALCEA

University of Texas at Dallas
Richardson, Texas, 75083-0688
rada@utdallas.edu

Abstract

The IRSLO (Information Retrieval using Semantic and Lexical Operators) project aims at integrating semantic and lexical information into the retrieval process, in order to overcome some of the impediments currently encountered with today's information retrieval systems. This paper introduces the semantic wildcard, one of the most powerful operators implemented in IRSLO, which allows for searches along general-specific lines. The semantic wildcard, denoted with #, acts in a manner similar with the lexical wildcard, but at semantic levels, enabling the retrieval of subsumed concepts. For instance, a search for *animal#* will match any concept that is of type *animal*, including *dog*, *goat* and so forth, thereby going beyond the explicit knowledge stated in texts. This operator, together with a lexical locality operator that enables the retrieval of paragraphs rather than entire documents, have been both implemented in the IRSLO system and tested on requests of information run against an index of 130,000 documents. Significant improvement was observed over classic keyword-based retrieval systems in terms of precision, recall and success rate.

1. Introduction

As the amount of information continues to increase, there must be new ways to retrieve and deliver information. Information is of no use if it cannot be located and the key to information location is a retrieval system. Traditionally, information retrieval systems use keywords for indexing and retrieving documents. These systems end up retrieving a lot of irrelevant information along with some useful information that the query/question was intended to elicit. Moreover, implicit knowledge makes often the bridge between a question and a document, and classic retrieval systems do not have the capability of going beyond explicit knowledge embedded in texts, thereby missing the answers to such queries.

To overcome some of the impediments currently encountered with today's information retrieval systems, we have started the IRSLO (Information Retrieval using Semantic and Lexical Operators) project that aims at integrating semantic and lexical information into the retrieval process, to the end of obtaining improved precision and recall. This paper introduces the *semantic wildcard*, one of the most powerful operators implemented in IRSLO.

Users' information needs are most of the times expressed along general-specific lines, and this paper provides analytical support towards this fact. *What sport*, *What animal*, *What body part*, are all examples of question types that require implicit knowledge about what constitutes a *sport*, *animal*, or *body-part*. The *semantic wildcard*, denoted with #, is designed to retrieve subsumed concepts. For instance, a search for *animal#* will match any concept that is of type *animal*, thereby going beyond the explicit knowledge stated in texts.

The *semantic wildcard*, together with a lexical locality operator previously introduced that enables the retrieval of paragraphs rather than entire documents (Mihalcea, 1999), were implemented in the IRSLO system and tested on requests of information run against an index of 130,000 documents. Significant improvement was observed over classic retrieval systems, in terms of precision, recall and success rate.

The paper is organized as follows. First, we present an analysis of questions asked by real time users, bringing evidence towards the fact that information need is most of the times expressed along general-specific lines. Next, we show how a novel encoding scheme - referred to as *DD-encoding* - can be applied to WordNet, in order to exploit the general-specific relations encoded in this semantic net. We then present the architecture of IRSLO, with emphasis on the *semantic wildcard* operator and the *paragraph operator*, together with experiments, results and walk through examples.

2. Defining Information Need

In order to define users' information need and assess the role that may be played by semantics in an information retrieval environment, we have performed a qualitative and quantitative analysis of information requests expressed by users in the form of natural language questions. Two sets of data are used during the experiments: (1) the Excite question log, for a total of 68,631 questions asked by the users of a search engine and (2) the TREC-8, TREC-9 and TREC-10 questions, for a total of 1,393 questions.

The noisy Excite log was cleaned up with two filters. First, we extracted only those lines containing one of the keywords *Where*, *When*, *What*, *Which*, *Why*, *Who*, *How*, *Why* or *Name*. Next, we eliminated the lines containing the phrase *"find information"* to avoid the bias towards Web searching questions. [1]

From the total of 25,272 Excite *What* questions[2] we have randomly selected a subset of 5,000 questions that were manually analyzed and classified. The decision of what question type to assign to a particular question was

[1] To our knowledge, only one other large scale question analysis is mentioned in the literature (Hovy et al., 2001).

[2] We emphasize the experiments involving *What* questions, since they provide the largest coverage and are considered to be the most ambiguous types of questions. Similar analyses were performed for the other types of questions, but are not reported here due to lack of space.

merely based on the possibility of implementing a procedure that would make use of this question type in the process of finding relevant information. For instance, a question like *What does Acupril treat?* expects a DISEASE as answer, which is doable in the sense that an ontology like WordNet does have a disease node with pointers to a large number of disease names. On the other hand, *What about this Synthyroid class action?* does not require a specific answer, but rather information related to a topic, and therefore no question type is assigned to this question (the type NONE is used instead). For the entire set of 5,000 questions, 361 categories are extracted.

2.1. Quantitative Analysis

To the end of observing the behavior and learning rate associated with question types, subsets of different sizes were created and the number of question types was determined for each subset. The measurements were performed using a 10-fold cross validation scheme on randomly selected samples of data.

Figure 1 plots the distribution of question types with respect to the subset size. It turns out that the number of question types grows sublinearly with the number of questions. Moreover, we noticed a behavior of the curve similar with *Heaps' Law* (Heaps, 1978), which relates the number of words in a text with the text size. *Heaps' Law* states that the size of the vocabulary for a text of size n is $V = Kn^{\beta} = O(n^{\beta})$.

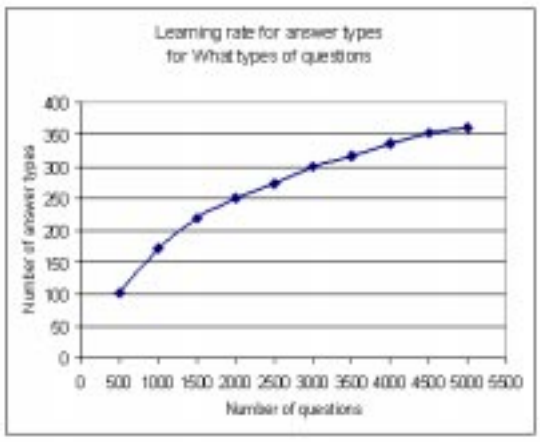


Figure 1: Number of question types vs. number of questions for *What* questions in the Excite log.

Denoting the number of question types with T_q and the number of questions with N_q, it follows:

$$\mathrm{T}_q = K{N_q}^{\beta} \quad (1)$$

The equation is solved by taking the log in both sides. For the Excite *What* set, it results a value of $K = 5.18$, respectively $\beta = 0.50$. The values of the two parameters are changed in the TREC *What* set: $K = 3.89$ and $\beta = 0.54$, which illustrates the difference in question types distribution for the uniform TREC set versus the noisy Excite set.

This is an interesting result, as it defines the behavior of question types with respect to the number of questions. Moreover, it gives us the capability of making estimates on what is the expected number of question types for N_q given questions. For instance, 10,000 questions will result in about 518 question types, 100,000 in about 1,638 question types, and so forth.

2.2. Qualitative Analysis

The qualitative analysis brings evidence for the organization of question types in semantic hierarchies, and supports the idea of incorporating semantics into information retrieval.

An analysis of the questions benchmarks suggested that the majority of question types are found in a general-specific (ISA) relation. This hypothesis is sustained by empirical evidence. We classified the questions into four categories as listed in Table 1[3]. It turns out that on average about 60% of the questions are clear general-specific questions. It is debatable whether or not the DEFINITION types of questions can be classified as general-specific questions or not. It is often the case that a definition requires a more general concept to explain an unknown entity (Prager et al., 2001), and therefore it could be considered as a general-specific information request. Under this hypothesis, it results an average of 80% of information requests being expressed along general-specific lines.

Information type	Frequency
Excite questions	
GENERAL-SPECIFIC	54.6%
DEFINITION	19.6%
NONE	14.8%
OTHER	10.8%
TREC questions	
GENERAL-SPECIFIC	65.0%
DEFINITION	20.9%
NONE	6.6%
OTHER	7.4%

Table 1: Information requests along general-specific lines

Figure 2 shows examples of annotated questions extracted from the Excite log, mapped on an *animal* hierarchy of question types.

The conclusion of these experiments is that the majority of information requests are expressed along general-specific lines, and therefore a semantic based retrieval system that exploits these relations would possibly increase the quality of the information retrieved. This idea was also expressed by (Berners-Lee et al., 2001) in the context of Semantic Web.

3. Conversion of WordNet to DD-encoding

On the one side, we have the users' information need expressed most of the times as a general-specific request.

[3]The OTHER category includes questions that require an answer that cannot be obtained by following a general-specific line. Examples of such question types are CAUSE, EFFECT, QUOTE|ALBUM, QUOTE|MOVIE, WORD-TRANSLATION, etc.

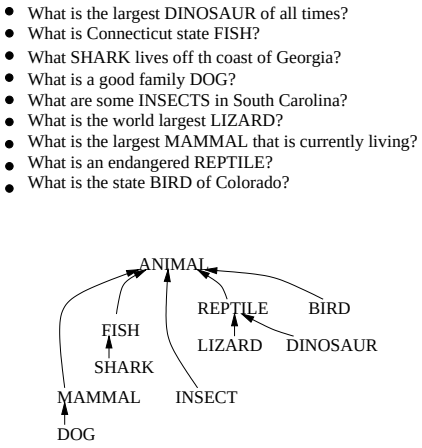


Figure 2: Question types mapped onto the *animal* hierarchy.

On the other side, we have WordNet (Miller, 1995) as the largest general purpose semantic network available today, which encodes about 86,605 general-specific (ISA) relations. We want to exploit as much as possible the semantic network structure of WordNet. To this end, we propose in this section a new encoding to be used for WordNet entries that would enable more efficient semantic searches. The so called *DD-encoding* was inspired by the Dewey Decimal code scheme used by librarians.

There are many times when keywords in a query are used with "generic" meanings and they are intended as representatives for entire categories of objects. *Foxes eat hens* is a statement that can be evaluated as a good match for *Animals eat meat*. Unfortunately, with current indexing and retrieval techniques this is not possible, unless both *animal* and *meat* are expanded with their subsumed concepts, which may sometimes become a tedious process. For this particular example, WordNet defines 7,980 concepts underneath *animal*, and there are 199 entries that inherit from *meat*, and therefore we end up with more than 1,500,000 (7,980 x 199) queries to cover the entire range of possibilities. Alternatively, if boolean queries are allowed and the OR operator is available, a query with 8,179 (7,980 + 199) terms can be used. None of these solutions seems acceptable and this is why none of them have been used so far.

We would like to find a way such that *fox* matches *animal* and we propose the employment of matching codes as an elegant solution to accomplish this task.

Finding the means that would allow for this type of matches is a problem of central interest for retrieval applications, as most information requests are expressed along general-specific lines. We want to retrieve documents containing *cat* in return to a search for *animal*, and retrieve *dachshund* and do not retrieve *cat* as the result of a search for *dog*.

To enable this type of general-specific searches and at the same time take advantage of the semantic structure already encoded in WordNet, we propose the employment of a codification scheme similar with the one used in librarian systems, and associate a code to each entry in WordNet.

The role of this code is to make evident to an external tool, such as an indexing or retrieval process, the relation that exists between inter-connected concepts. No information can be drawn from the simple reading of the *animal* and *dog* strings. Things are completely different when we look at *13.1* and *13.1.7*: the *implicit* relation between the two tokens has now been turned into an *explicit* one.

A code is assigned to each WordNet entry such that it replicates its parent code, and adds a unique identifier. For instance, if *animal* has code *13.1*, then *chordate*, which is a directly subsumed concept, has code *13.1.29*, *vertebrate* has code *13.1.29.3*, and so forth. Figure 3 illustrates a snapshot from the noun WordNet hierarchy and shows the *DD-codes* attached to each node. This encoding creates the grounds for matching at semantic levels in a manner similar with the lexical matches already employed by several information retrieval systems.

To our knowledge, this is a completely new approach taken towards the goal of making possible searches at semantic levels. The idea underneath this encoding is very simple but it allows for a powerful operator: the *semantic wildcard*.

3.1. Technical Issues

There are several implementation issues encountered during WordNet transformation, and we shall address them in this section.

Specifically, the new encoding is created using the following algorithm:

---

1. Start with the top of WordNet hierarchies. For each top, load its hyponyms, and for each hyponym go to step 2.
2. Execute the following steps:
2.1. Assign to the current synset the DD-code of its parent plus an unique identifier that is generated as a number in a successive series.
2.2. If the current synset has been already assigned a DD-code, then generate a special link between its parent and the current synset itself.
2.3. Load all hyponyms of current synset and go to step 2.

The algorithm performs a recursive traversal of the entire WordNet hierarchy and generates codes. A code is associated with a synset, and we created a list of pairs containing a synset offset (the current WordNet encoding) and a *DD-code*.

It is worth mentioning the case of multiple inheritance, handled by the Dewey classification system as an addition made for a particular category. For instance, 675+678 means *leather and rubber*. This solution is not satisfactory for our purpose, since it may result in very long codes. Instead, a list of *special links* (generated in step 2b) is created, containing all the links between a *second parent* and a child. For example, if *house* inherits from both *domicile* and *building*, we have the code 1.2.1.32.12.23 for *house*, 1.2.1.32.28.6 for *domicile* and 1.2.1.32.12 for *building*, and in addition a special link is generated to indicate that *domicile* is the parent of *house* even if no direct matching can be performed.

For the entire noun hierarchy in WordNet, 74,488 *DD-codes* were generated. In addition, 4,280 multiple inheritance links were created. The average length of a code is 16 characters. Given the fact that disk space is a cheap re-

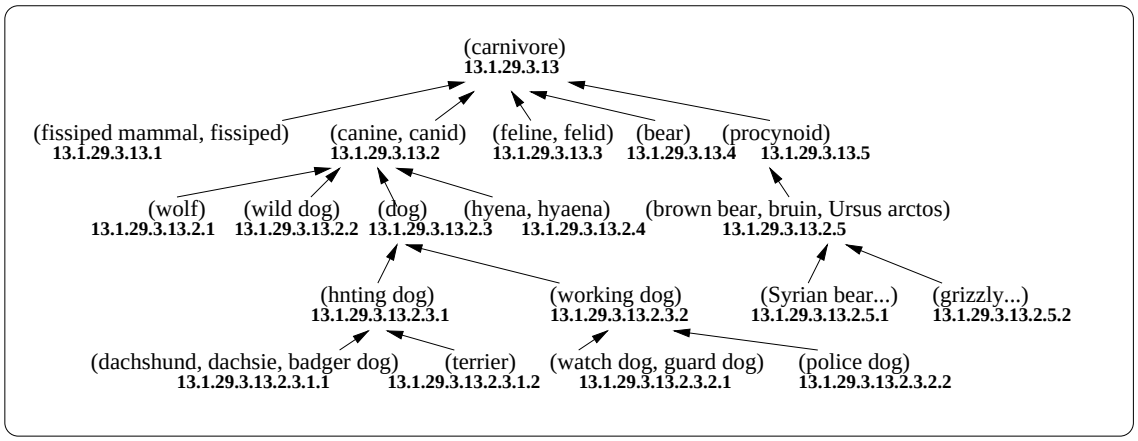


Figure 3: *DD-codes* assigned to a sample of the WordNet hierarchy

source, the length of the codes does not represent a real disadvantage of the proposed approach. Moreover, one should take into consideration that no optimizations were sought in the process of code generation. A simple strategy, like the usage of all 256 ASCII characters instead of using only the 1-9 digits, can shorten significantly the length of the codes (e.g. 1.2.1.32.12.23 changes into 1.2.1.z.b.f). Approaches like Huffman code or other compression methods can be as well exploited for this purpose, but we will not consider these issues here.

4. The IRSLO System

Our improved semantic based information retrieval system comprises the same main components as found in any other retrieval system.

4.1. Question/Query Processing

This stage usually includes a keyword selection process. It may sometimes imply keyword stemming or other processing, and in most cases keywords to be employed in the retrieval stage are selected based on weights, frequencies and stop-words lists.

In IRSLO, we start this stage with a simple tokenization and part of speech tagging using Brill tagger (Brill, 1995). Next, collocations are identified based on WordNet definitions. We also identify the baseform of each word.

Depending on the notation employed by the user, we distinguish three keyword types. (1) Words with a semantic wildcard, denoted with #. (2) Words to be searched by their *DD-code*, denoted with @ (synonymy marker). (3) Words with no special notation, to be sought in the index in their given form. By default, we assume a # assigned to the answer type word, and no other notation for the rest of the words. All words that are denoted with # or @ are passed on to a word sense disambiguation component that solves their semantic ambiguity. Alternatively, this step can be skipped and a default sense of one with respect to WordNet is assigned, with reasonable precision (over 75% as measured on SemCor). The results reported in this paper are based on a simplified implementation that considers the second alternative. Next, *DD-codes* are assigned to words in text and subsequently used in the retrieval process. *DD-codes* are currently assigned only to nouns, considered to be the most informative words. See section 3. for more details regarding *DD-encoding*.

We also face the task of identifying relevant keywords to be included in a query. Extensive analysis of keywords identification was previously reported in (Pasca, 2001). We use a simplified keywords identification procedure, based on the following rules:

1. Use all proper nouns and quoted words.
2. Use all nouns.
3. Use all adjectives in superlative form.
4. Use all numbers (cardinals).
5. If more than 200 documents are returned, use the adjectives modifying the first noun phrase.
6. If no documents are returned, drop the nouns acting as modifiers. Particular attention is paid to abstract nouns, such as type, kind, name, where the importance of the roles played by a head and a modifier in a noun phrase are interchanged.

Any of these keywords may be expressed using its corresponding *DD-code*. The answer type word is also important. It practically denotes the type of information sought, whether is a *country*, an *animal*, a *fish*, etc. We use a simple approach that selects the answer type as the head of the first noun phrase. There are few exceptions from this rule, consisting of the cases where the head is an abstract noun like *name*, *type*, *variety* and so forth, and in such cases we select its modifier. If the answer detected is of a generic type, such as *person*, *location*, *organization*, then we replace it with the corresponding named entity tag. Otherwise, the answer type word is assigned a # semantic wildcard. Notice that the answer type selection process is invoked only if there is no word a priori denoted with #.

After all these processing steps, we end up with a query in IRSLO format. The words that were assigned a semantic wildcard # are now represented as *DD-code**. The words with a synonymy marker are simply replaced with their *DD-code* (thereby allowing for the retrieval of synonym words in addition to the word itself). The other words are replaced with their baseform. See Section 5.4. for representation examples.

4.2. Document Processing

Typically, documents are simply tokenized and terms are extracted, in preparation for the indexing phase. Optionally, stop-words are eliminated and words are stemmed prior to indexing.

In IRSLO, documents are processed following similar steps to question processing. First, the text is tokenized and part of speech tagged. We have an additional component that involves named entity recognition (Lin, 1994). Next, we identify compound words, apply a disambiguation algorithm or, alternatively, assign to each word its default sense from WordNet. Finally we assign to each noun its corresponding *DD-code*.

At this stage, we also identify paragraphs and store them as one paragraph per line. This helps improving efficiency during paragraph retrieval.

4.3. Indexing and Retrieval

The indexing process is not different in any ways with respect to a classic information retrieval system. A TF/IDF weight is assigned to each term. We index complex terms, including the *DD-codes* attached to each noun and the named entity tags, when available. No additional stemming or stop-words elimination is performed. The retrieval system allows for flexible searches, including regular expressions. Based on *DD-codes*, we have the capability of using the *semantic wildcard* operator, in addition to the lexical wildcard. We also have the capability of retrieving named entities of a certain type (e.g. perform a search for *person*). Moreover, we allow for boolean operators and for the new *paragraph operator* for a more focused search. Documents are ranked using the TF/IDF weight associated with each keyword.

5. Experiments with IRSLO

This section focuses on the application of the *semantic wildcard* and *paragraph operator* within the IRSLO system. First, the semantic wildcard enables searches for information along general-specific lines. Second, the paragraph indexing component limits the scope of keywords search to a single paragraph, rather than an entire document.

5.1. Experimental Setup

Several standard text collections are made available through the Information Retrieval community. For our experiments, we have selected the *L.A. Times* collection, which includes a fairly large number of documents. There are more than 130,000 documents adding up to 500MB of text. *L.A. Times* is part of the TREC (Text REtrieval Conference) collections.

The main advantage of standard text collections is the fact that question sets and relevance judgments are usually provided in association with the document collection.

About 1,393 questions have been released during the TREC-8, TREC-9 and TREC-10 Q&A TREC competitions. Relevance judgments are provided for the first two competitions, i.e. for 893 questions. From the 893 questions, we selected only the *What* type of questions, as being the most ambiguous types of questions and the best candidates for the semantic wildcard operator. Subsequently, we identified those questions known to have an answer in the *L.A. Times* collection[4], and out of these 75 questions were randomly selected for further tests.

For this question set, we have the knowledge about the information expected in response to each question (answer patterns provided by the TREC community). We also have a list of *docid*-s pointing to documents containing the answer for each question (list of documents judged to contain a correct answer by TREC assessors). This information helps us measure *precision* and *recall*.

5.2. Evaluating Retrieval Effectiveness

A common methodology in evaluating information retrieval systems consists in measuring *precision* and *recall*. *Precision* is defined as the number of relevant documents retrieved over the total number of documents retrieved. *Recall* is defined as the number of relevant documents retrieved over the total number of relevant documents found in the collection. Additionally, the *F-measure* proposed in (Van Rijsbergen, 1979) provides the means for combining recall and precision into one single formula, using relative weights.

$$F_{measure} = \frac{(\beta^2 + 1.0) * P * R}{(\beta^2 * P) + R}$$

where P is precision, R is recall and β is the relative importance given to recall over precision. During the system evaluations reported here, we considered both precision and recall of equal importance, and therefore β is set to 1.

Moreover, we employ the *success rate* measure (Woods, 1997) as an indicative of how many questions were answered by the system. The *success rate* for a question/query is 1 if relevant documents/answers are found, and 0 otherwise.

Finally, we evaluate IRSLO results using the TREC Q&A score, with a different mark assigned to an answer depending on its position within the final rank. A correct answer on the first position results in a maximum score of 1.00. The second position gets 0.50, the third position is scored with 0.33, the fourth with 0.25 and the fifth and last one acceptable receives 0.20 points.

5.3. Experiments

Three types of experiments were performed, to evaluate the performance of the new *semantic wildcard* and *paragraph operator*.

Experiment 1. Extract the keywords[5] from each question and run the queries formed in this way against a classic index created with the *L.A. Times* collection. The purpose of this experiment is to simulate classic keyword-based retrieval systems. The ranking is provided through a TF/IDF weighting scheme.

Experiment 2. Extract the keywords from each question and run the queries against the paragraph index. In paragraph

[4]The set of 893 questions was devised to ensure an answer in the entire TREC collection, including 2.5GB of text in addition to the *LA Times* collection that we employ in our experiments

[5]See Section 4.1. for the keywords selection procedure

indexing, we use a boolean model that includes the *paragraph operator*, plus a measure that determines the closeness among keywords to rank the paragraphs.

Experiment 3. Again, extract keywords from questions and run them against the paragraph index. Additionally, we allow the *semantic wildcard* (including named entity tags) to be specified in the keywords.

The results of experiments 1 and 2 are compared, to show the power of paragraph indexing. Experiments 2 and 3 provide comparative results to support the use of semantics, specifically the *semantic wildcard*.

The first experiment represents a classic keyword-based information retrieval run, and therefore we evaluate it in terms of *precision*, *recall* and *F-measure*. The second and third experiments are also evaluated in terms of *precision*, *recall* and *F-measure*. Additionally, we use the *success rate* and *TREC score*.

5.4. Walk-through Examples

This section gives several running examples of the IRSLO system, using the *semantic wildcard* and *paragraph operator*.

Example 1. What is the brightest star visible from Earth?
Relevant paragraph. In the year 296036 , Voyager 2 will make its closest approach to Sirius , the brightest star visible from Earth .
Comments. The query formed in this case is star# AND bright AND Earth. Only two answers are found by the system, and the one listed above, which is the correct one, is ranked on the first position. Sirius is defined in WordNet as a star, and consequently was annotated as such in the text.

Example 2. What kind of sports team is the Buffalo Sabres?
Relevant paragraph. Another religious broadcasting company , Tri - State Christian TV Inc. of Marion , Ill. , which was set up with the help of loan guarantees from Trinity , announced recently that it has purchased WNYB Channel 49 in Buffalo , N.Y. , from the Buffalo Sabres hockey team for $2.5 million .
Comments. The query employed is team# AND Buffalo AND Sabres. The original query team# AND sport AND Buffalo AND Sabres did not return any answers, and consequently the back off scheme was invoked and dropped noun modifiers. A total of six paragraphs are found in return to this question, all of them correct.

Example 3. What U.S. Government agency registers trademarks?
Relevant paragraph. After your application arrives at the Patent Office , it is turned over to an attorney who determines whether there is anything " confusingly similar "between your trademark and others [...]
Comments. Patent Office is a type of Government agency, and therefore the query U.S. AND government_agency# AND trademark leads to the correct answer.

Example 4. What cancer is commonly associated with AIDS?
Relevant paragraph. A team of transplant specialists at City of Hope National Medical Center in Duarte is among several groups nationwide that plan to test the experimental procedure on a small number of patients with AIDS - related lymphomas , or tumors of the lymph nodes .
Comments. The query employed is cancer# AND AIDS. The answer was found at rank 4, and it seems that none of the teams in the TREC competition identified this answer, because there is no direct reference in the text to cancer, but only a hidden relation from lymphomas to cancer. Our semantic model has the capacity to detect such non-explicit relations.

5.5. Results

Tests were performed using the benchmark of 75 questions. For each question, we run three experiments, as mentioned earlier. (1) Keyword-based information retrieval using a TF/IDF scheme. (2) Paragraph indexing and retrieval (i.e. enable the paragraph operator). (3) An experiment that involves both paragraph operator and semantic wildcard.

Precision, *recall* and *F-measure* are determined for all these experiments. We have also determined *success rate* and *TREC score*.

Ten sample requests of information are presented below, with their evaluations shown in Table 2. The following notations are used: P = *precision*, R = *recall*, F = *F-measure*, SR = *Success Rate*, TS = *TREC score*.

1. What American composer wrote the music for "West Side Story"?
2. What U.S. Government agency registers trademarks?
3. What U.S. state's motto is "Live free or Die"?
4. What actor first portrayed James Bond?
5. What animal do buffalo wings come from?
6. What cancer is commonly associated with AIDS?
7. What city does McCarren Airport serve?
8. What instrument does Ray Charles play?
9. What is the population of Japan?
10. What is the tallest building in Japan?

Cumulative results for all 75 questions are compared in Table 2. It turns out that the *F-measure* doubles when paragraph indexing is used with respect to document indexing, with increased *precision* and lower *recall*, as expected. The *success rate* is determined for the second and third experiments to evaluate the effect of the *semantic wildcard* over simple paragraph indexing, and an increase of 17% is observed. As of the *TREC score*, the additional use of semantics brings a gain of 34% with respect to simple paragraph indexing.

These results are very encouraging, and in agreement with the suggestions made in (Light et al., 2002) that query expansion and semantic relations are essential for increased performance, for information retrieval in general and Q&A systems in particular.

6. Related Work

Significant work has been performed in the field of semantics applied to information retrieval. The most important directions include: (1) query expansion (Voorhees, 1998), (2) phrase indexing (Strzalkowski et al., 1996), (3) conceptual indexing (Woods, 1997), (4) semantic indexing (Sussna, 1993), (Krovetz, 1997). In addition, the Semantic Web is a new field that considers the use of semantics for Web applications (Berners-Lee et al., 2001).

7. Conclusion

This paper has introduced the *semantic wildcard*, a novel operator that enables the use of semantics in information retrieval applications. The *semantic wildcard*, together with the new *paragraph operator*, were implemented in the IRSLO system. Experiments were performed on a collection of 130,000 documents with 75 *What*-questions extracted from the questions released during TREC competitions. Three experiments were performed. (1) One that

Question number	Experiment												
	1. Classic IR			2. Par.op.					3. Sem.wildcard + par.op.				
	P	R	F	P	R	F	SR	TS	P	R	F	SR	TS
1	0.14	0.21	0.17	0.50	0.07	0.12	1	1.00	0.75	0.86	0.80	1	1.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	1.00	1.00	1.00	1	1.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	1.00	0.67	0.80	1	1.00
4	0.25	0.44	0.32	0.43	0.17	0.24	1	1.00	0.16	1.00	0.27	1	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	0.43	1.00	0.60	1	0.33
6	0.08	0.84	0.14	0.03	1.00	0.03	1	0.00	0.37	0.74	0.49	1	0.25
7	1.00	1.00	1.00	1.00	1.00	1.00	1	1.00	1.00	1.00	1.00	1	1.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	0.38	1.00	0.55	1	0.50
9	0.03	1.00	0.05	0.04	0.50	0.07	1	0.00	0.08	0.33	0.13	1	1.00
10	0.03	0.50	0.06	0.40	0.50	0.44	1	1.00	1.00	1.00	1.00	1	1.00

Table 2: Precision, recall, F-measure, success rate and TREC score for 10 sample requests of information

Measure	Experiment		
	1. Classic IR	2. Par.op.	3. Sem.wildcard. + par.op.
Precision	0.05	0.12	0.12
Recall	0.66	0.57	0.61
F-measure	0.092	0.19	0.20
Success rate	-	66.0%	77.3%
TREC score	-	43.4%	58.3%

Table 3: Comparative results for (1) keyword-based information retrieval (2) paragraph operator and (3) paragraph operator + semantic wildcard

simulates classic keyword-based information retrieval with a TF/IDF weighting scheme. (2) A second experiment that implements the *paragraph operator*. (3) Finally, a third experiment where both *semantic wildcard* and *paragraph operator* are employed. Various measures were used to evaluate the performance attained during these experiments, and all measures have proved the efficiency of our *semantic wildcard* operator, respectively the *paragraph operator*, over keyword-based retrieval techniques. As a follow-up analysis, it would be interesting to determine the *min* and *max* bounds proposed in (Light et al., 2002) for the precision achievable on a question set when the semantic wildcard is enabled.

8. References

T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American*, 1(501), May.

E. Brill. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December.

H.S. Heaps. 1978. *Information Retrieval, Computational and Theoretical Aspects*. Academic Press.

E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the Human Language Technology Conference, HLT 2001*, San Diego, CA.

R. Krovetz. 1997. Homonymy and polysemy in information retrieval. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 72–79.

M. Light, G.S. Mann, E. Riloff, and E. Breck. 2002. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering (forthcoming)*.

D. Lin. 1994. Principar - an efficient, broad-coverage, principle-based parser. In *In Proceedings of the Fifteenth International Conference on Computational Linguistics COLING-ACL '94*, pages 42–48, Kyoto, Japan.

R. Mihalcea. 1999. Word sense disambiguation and its application to the Internet search. Master's thesis, Southern Methodist University.

G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

M. Pasca. 2001. *High performance question answering from large text collections*. Ph.D. thesis, Southern Methodist University.

J. Prager, D. Radev, and K. Czuba. 2001. Answering what-is questions by virtual annotation. In *Proceedings of the Human Language Technology Conference, HLT 2001*, San Diego, CA.

T. Strzalkowski, L. Guthrie, J. Karigren, J. Leistensnider, F. Lin, J. Perez-Caballo, T. Straszheim, J. Wang, and J. Wilding. 1996. Natural language information retrieval, TREC-5 report. In *Proceedings of the 5th Text Retrieval Conference (TREC-5)*, pages 291–314, Gaithersburg, Maryland, November.

M. Sussna. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management CIKM '93*, pages 67–74, Washington, November.

C.J. Van Rijsbergen. 1979. *Information Retrieval*. London: Butterworths. available on-line at http://www.dcs.gla.ac.uk/Keith/Preface.html.

E.M. Voorhees. 1998. Using WordNet for text retrieval. In *WordNet, An Electronic Lexical Database*, pages 285–303. The MIT Press.

W.A. Woods. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, April. available online at: *http://www.sun.com/ research/techrep/ 1997/abstract-61.html*.

Statistical Query Disambiguation, Translation and Expansion in Cross-Language Information Retrieval

Fatiha Sadat*, Akira Maeda†, Masatoshi Yoshikawa‡*, Shunsuke Uemura*

* Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)
8916-5 Takayama, Ikoma. Nara 630-0101. Japan

† CREST, Japan Science and Technology Corporation (JST)
‡ National Institute of Informatics (NII)
{fatia-s, aki-mae, yosikawa, uemura}@is.aist-nara.ac.jp

Abstract

Query expansion is considered as one of the most important methods in improving the effectiveness of information retrieval. By combining query expansion with dictionary-based translation and statistics-based disambiguation, in order to overcome query terms ambiguity, information retrieval should become much more efficient. In the present paper, we focus on query terms disambiguation via, a combined statistical method both before and after translation, in order to avoid source language ambiguity as well as incorrect selection of target translations. Query expansion techniques through relevance feedback were performed prior to either the first or the second disambiguation processes. We tested the effectiveness of the proposed combined method, by an application to a French-English Information Retrieval. Experiments involving TREC data collection revealed the proposed disambiguation and expansion methods to be highly effective.

1. Introduction

In recent years, the number of studies concerning Cross-Language Information Retrieval (CLIR) has grown rapidly, due to the increased availability of linguistic resources for research. Cross-Language Information Retrieval consists of providing a query in one language and searching document collections in one or more languages. Therefore, a translation form is required. In the present paper, we focus on query translation, disambiguation and expansion in order to improve the effectiveness of information retrieval through various combinations of these methods. First, we are interested to find retrieval methods that are capable of performing across languages and which do not rely on scarce resources such as parallel corpora. Bilingual Machine Readable-Dictionaries (MRDs), more prevalent than parallel texts, appear to be a good alternative. However, simple translations tend to be ambiguous and yield poor results. A combination that includes a statistical approach for a disambiguation can significantly reduce errors associated with *polysemy*[1] in dictionary translation. In addition, automatic query expansion, which has been known to be among the most important methods in overcoming the word mismatch problem in information retrieval, is also considered. As an assumption to reduce the effect of ambiguity and errors that a dictionary-based method would cause, a combined statistical disambiguation method is performed both prior to and after translation. Although, the proposed information retrieval system is general across languages in information retrieval, we conducted experiments and evaluations concerning French-English information retrieval.

The remainder of the present paper is organized as follows. Section 2 provides a brief overview of related works. Both dictionary-based and the proposed disambiguation methods are described in Section 3. A combination involving query expansion is described in Section 4. Evaluation and discussion of the experiments of the present study are presented in Section 5. Section 6 involves Word Sense Disambiguation and Section 7 describes the conclusion of the present paper.

2. Related Research in CLIR

The potential of knowledge-based technology has led to increasing interest in CLIR. The query translation of an automatic MRD, on its own, has been found to lead to a drop in effectiveness of 40-60 % compared to monolingual retrieval (Hull and Grefenstette, 1996; Ballesteros and Croft, 1997). Previous studies have used MRDs successfully, for query translation and information retrieval (Yamabana et al., 1996; Ballesteros and Croft, 1997; Hull and Grefenstette, 1996). However, two factors limit the performance of this approach. The first is that many words do not have a unique translation and sometimes the alternate translations have very different meanings (*homonymy and polysemy).* The fact that a single word may have more than one sense is called *ambiguity.* Translation ambiguity significantly exacerbates the problem in CLIR (Oard, 1997). Most of the previously proposed disambiguation strategies rely on statistical approaches, but without considering ranking or selection of source query terms, which affect directly the selection of target translations. The second challenge is that dictionary may lack some terms that are essential for a correct interpretation of the query. In the present study, we propose the concept of the combined statistical disambiguation technique, applied prior to and after dictionary translation to solve lexical semantic ambiguity. In addition, a monolingual thesaurus is introduced to overcome bilingual dictionary limitation. Automatic query expansion through relevance feedback, which has been used extensively to improve the effectiveness of an information retrieval (Ballesteros and Croft, 1997; Loupy et al., 1998), is considered. Selection of expansion terms was performed through various means. In the present study, we use a ranking factor to select the best expansion terms-those related to all source query terms, rather than to just one query term.

[1] Polysemy is a word, which has more than one meaning.

3. Translation/Disambiguation in CLIR

There are two types of lexical semantic ambiguity with which a machine translation system must contend: there is ambiguity in the source language where the meaning of a word is not immediately apparent but also ambiguity in the target language when a word is not ambiguous in the source language but it has two or more possible translations (Hutchins and Sommers, 1992). In the present research, query translation/disambiguation phases are performed after a simple *stemming* process of query terms, replacing each term with its inflectional root and each verb with its infinitive form, as well removing most plural word forms, stop words and stop phrases. Three primary tasks are completed using the translation/disambiguation module. First, an *organization of source query terms*, which is considered key to the success of the disambiguation process, will select best pairs of source query terms. Next a *term-by-term translation* using the dictionary-based method (Sadat et al., 2001), where each term or phrase in the query is replaced by a list of its possible translations, is completed. Missing words in the dictionary, which are essential for the correct interpretation of the query. This may occur either because the query deals with a technical topic, which is outside the scope of the dictionary or because the user has entered some form of abbreviations or slang, which is not included in the dictionary (Oard, 1997). To solve this problem, an automatic *compensation* is introduced, via synonym dictionary or existing thesaurus in the concerned language. This case requires an extra step to look up the query term in the thesaurus or synonym dictionary, find equivalent terms or synonyms of the targeted source term, thus performing a query translation. In addition, short queries of one term are concerned by this phase. The third task, *disambiguation of target translations*, selects best translations related to each source query term. Finally, documents are retrieved in target language.

Figure 1 shows thee overall design of the proposed information retrieval system. Query expansion will be applied prior to and/or after the translation/disambiguation process. Among the proposed expansion strategies are, relevance feedback and thesaurus-based expansion, which could be interactive or automatic.

3.1. Organization of Source Query Terms

All possible combinations of source query terms are constructed and ranked depending on their mutual co-occurrence in a training corpus. A type of statistical process called *co-occurrence tendency* (Maeda et al., 2000; Sadat et al., 2001) can be used to accomplish this task. Methods such as Mutual Information MI (Church and Hanks, 1990), the Log-Likelihood Ratio LLR (Dunning, 1993), the Modified Dice Coefficient or Gale's method (Gale and Church, 1991) are all candidates to the co-occurrence tendency.

3.2. Co-occurrence Tendency

If two elements often co-occur in the corpus, then these elements have a high probability of being the best translations among the candidates for the query terms. The selection of pairs of source query terms to translate as well as the disambiguation of translation candidates in order to select target ones, is performed by applying one of the statistical methods based on co-occurrence tendency, as follows:

• *Mutual Information (MI)*

This estimation uses *mutual information* as a metric for significance of word co-occurrence tendency (Church and Hanks, 1990), as follows:

$$MI(w_1, w_2) = \log \frac{Prob(w_1, w_2)}{Prob(w_1)Prob(w_2)}$$

Here, $Prob(w_i)$ is the frequency of occurrence of word w_i divided by the size of the corpus N, and $Prob(w_i, w_j)$ is the frequency of occurrence of both w_i and w_j together in a fixed window size in a training corpus, divided by the size of the corpus N.

• *Log-Likelihood Ratio (LLR)*

The Log-Likelihood Ratio (Dunning, 1993) has been used in many researches. LLR is expressed as follows:

$$-2\log \lambda = K_{11}\log \frac{K_{11}N}{C_1R_1} + K_{12}\log \frac{K_{12}N}{C_1R_2} + K_{21}\log \frac{K_{21}N}{C_2R_1} + K_{22}\log \frac{K_{22}N}{C_2R_2}$$

Where, $C_1 = K_{11} + K_{12}$, $C_2 = K_{21} + K_{22}$, $R_1 = K_{11} + K_{21}$, $R_2 = K_{12} + K_{22}$, $N = K_{11} + K_{12} + K_{21} + K_{22}$,
K_{11} = *frequency of common occurences of word* w_i *and word* w_j , K_{12} = *corpus frequency of word* w_i - K_{11} ,
K_{21} = *corpus frequency of word* w_j - K_{11} , $K_{22} = N - K_{12} - K_{22}$.

3.3. Disambiguation of Target Translations

A word is *polysemous* if it has senses that are different but closely related. As a noun, for example, *right* can mean something that is morally acceptable, something that is factually correct, or one's entitlement. A two-terms disambiguation of translation candidates can be applied (Maeda et al., 2000; Sadat et al., 2001) is required, following a dictionary-based method. All source query terms are generated, weighed, ranked and translated for a disambiguation through co-occurrence tendency. The classical procedure for a *two-term disambiguation*, is described as follows:

1. Construct all possible combinations of pairs of terms, from the translation candidates.
2. Request the disambiguation module to obtain the co-occurrence tendencies. The window size is set to one paragraph of a text document rather than a fixed number of words.
3. Choose the translation, which shows the highest co-occurrence tendency, as the most appropriate.

As illustrated in Figure 2, the disambiguation procedure is used for two-term queries due to the computational cost (Maeda et al., 2000). In addition, the primary problem concerning long queries, involves the selection of pairs of terms, as well as the order for disambiguation. We propose and compare two methods for *n-term disambiguation,* for queries of two or more terms. The first method is based on a ranking of pairs of source query terms before the translation and disambiguation of target translations. The key concept in this step is to maintain the ranking order from the organization phase and perform translation and disambiguation starting from the most informative pair of source terms, i.e. a pair of source query terms having the highest co-occurrence tendency. Co-occurrence tendency is involved in both phases,

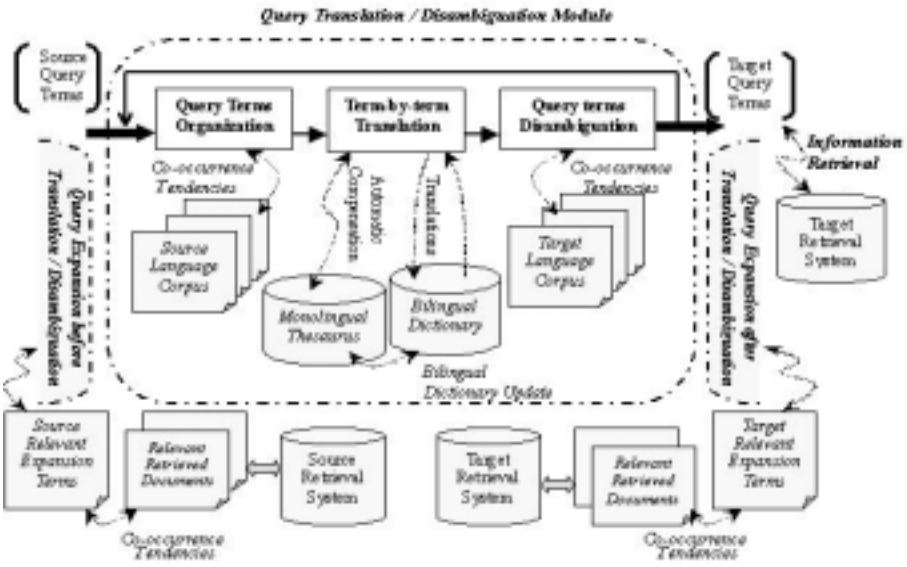


Figure 1: An overview of the Proposed Information Retrieval System
(In this research, source/target languages are French/English)

organization for source language and disambiguation for target language. The second method is based on a ranking of target translation candidates. These methods are described as follows: Suppose, Q represents a source query with n terms $\{s_1, s_2, \ldots, s_n\}$.

First Method: *(Ranking source query terms and disambiguation of target translations)*

1. Construct all possible combinations of terms of one source query: (s_1, s_2), (s_1, s_3), … (s_{n-1}, s_n).
2. Rank all combinations, according to their co-occurrence tendencies[2] toward highest values.
3. Select the combination (s_i, s_j), having the highest co-occurrence tendency, where at least one translation of the source terms has not yet been fixed.
4. Retrieve all related translations to this combination from the bilingual dictionary.
5. Apply a two-term disambiguation process to all possible translation candidates,
6. Fix the best target translations for this combination and discard the other translation candidates.
7. Go to the combination having the next highest co-occurrence tendency, and repeat steps 3 to 7 until every source query term's translation is fixed.

Second Method: *(Ranking and disambiguation of target translations)*

1. Retrieve all possible translation candidates for each source query term s_i, from the bilingual dictionary.
2. Construct sets of translations T_1 , T_2, …, T_n related to each source query term s_1 , s_2, …, s_n, and containing all possible translations for the concerned source term. For example, $T_i = \{t_{i1}, \ldots, t_{in}\}$ is the translation set for term s_i.
3. Construct all possible combinations of elements of different sets of translations. For example, (t_{11}, t_{21}), (t_{11} , t_{22}), …. (t_{ij} , t_{nk}),
4. Select the combination having the highest co-occurrence tendency[2].
5. Fix these target translations, for the related source terms and discard the other translation candidates.
6. Go to the next highest co-occurrence tendency and repeat step 4 through 6, until every source query term's translation is fixed.

Examples using the two proposed disambiguation methods are shown in Figures 3 and 4 for source English queries and target French translations.

4. Query Expansion in CLIR

Following the research reported by (Ballesteros and Croft, 1997) on the use of local feedback, the addition of terms that emphasize query concepts in the pre and post-translation phases improves both precision and recall. In the present study, we have proposed the combined automatic query expansion before and after translation through a relevance feedback. Original queries were modified, using judgments of the relevance of a few highly ranked documents, obtained by an initial retrieval, based on the presumption that those documents are relevant. However, query expansion must be handled very carefully. Simply selecting any expansion term from relevant retrieved documents could be risky. Therefore, our selection is based on the co-occurrence tendency in conjunction with all terms in the original query, rather than with just one query term. Assume that we have a query Q with n terms, $\{term_1 \ldots term_n\}$, then a ranking factor based on the co-occurrence frequency between each term in the query and an expansion term candidate, already extracted from the top retrieved relevant documents, is evaluated as:

$$Rank(expterm) = \sum_{i=1}^{n} co - occur(term_i, exp\ term)$$

[2] Co-occurrence tendency is based on one of the statistical methods: Mutual Information or Log-Likelihood Ratio, …

where, *co-occur(term$_i$, expterm)* represents the co-occurrence tendency between a query term *term$_i$* and the targeted expansion candidate *expterm*. *Co-occur(term$_i$, expterm)* can be evaluated by any estimation technique, such as mutual information or the log-likelihood ratio. All co-occurrence values were computed and then summed for all query terms (*i =1 to n*). An expansion candidate having the highest rank was then selected as an expansion term for the query Q. Note that the highest rank must be related to at least the maximum number of terms in the query, if not all query terms. Such expansion may involve several expansion candidates or just a subset of the expansion candidates.

5. Experiments and Evaluation

Experiments to evaluate the effectiveness of the two proposed disambiguation strategies, as well as the query expansion, were performed using an application of French-English information retrieval, i.e. French queries to retrieve English documents.

5.1. Linguistics Resources

Test Data: In the present study, we used test collection 1 from the TREC[3] data collection. Topics 63-150 were considered as English queries and were composed of several fields. Tags <num>, <dom>, <title>, <desc>, <smry>, <narr> and <con> denote topic number, domain, title, description, summary, narrative and concepts fields, respectively. Key terms contained in the title field <title> and description field <desc>, an average of 5.7 terms per query, were used to generate English queries. Original French queries were constructed by a native speaker, using manual translation.

Monolingual Corpora: The Canadian Hansard corpus (parliament debates) is a bilingual French-English parallel corpus, which contains more than 100 million words of English text as well as the corresponding French translations. In the present study, we used Hansard as a monolingual corpus for both French and English languages.

Bilingual Dictionary: COLLINS French-English dictionary was used for the translation of source queries.

Monolingual Thesaurus: EuroWordNet (Vossen, 1998) a lexical database was used to compensate, for possible limitations in the bilingual dictionary.

Stemmer and Stop Words: Stemming was performed using the English *Porter*[4] *Stemmer*. A special French stemming was developed and used in these experiments.

Retrieval System: The *SMART Information Retrieval System*[5] was used to retrieve both English and French documents. SMART is a vector model, which has been used in several studies concerning Cross-Language Information Retrieval.

5.2. Experiments and Results

A retrieval using original English/French queries was represented by *Mono_Fr/Mono_Eng* methods, respectively. We conducted two types of experiments. Those related to the query translation/disambiguation and those related to the query expansion before and/or after translation. Document retrieval was performed using original and constructed queries by the following methods. *All_Tr* is the result of using all possible translations for each source query term, obtained from the bilingual dictionary. *No_DIS* is the result of no disambiguation, which means selecting the first translation as the target translation for each source query term. We tested and evaluated two methods fulfilling the disambiguation of

[3] http://trec.nist.gov/data.html
[4] http://bogart.sip.ucm.es/cgi-bin/webstem/stem
[5] ftp://ftp.cs.cornell.edu/pub/smart

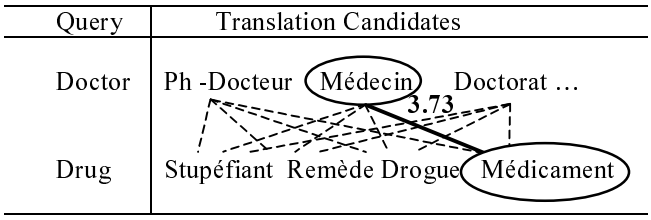


Figure 2: Two-Term Disambiguation Process

Highest co-occurrence tendencies for combinations of target translation candidates are as follows: (médecin, médicament), (médecin, remède), (médecin, drogue) …
Source French query: "doctor drug". Translated query to English: "médecin mèdicament".

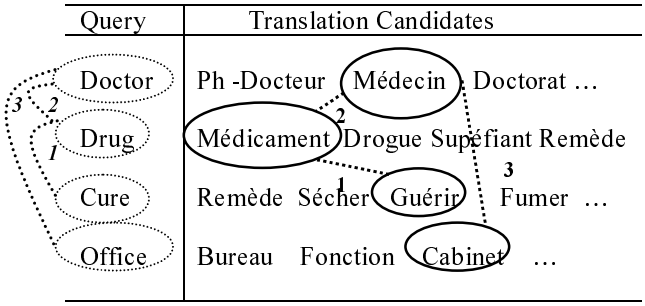


Figure 3: N-Term Disambiguation *(First Method)*: Ranking Source Query Terms and Disambiguation of Target Translations

Highest co-occurrence tendencies related to pairs of source query terms are as follows: (drug, cure), (doctor, drug), (doctor, office), (doctor, cure)…
Source French query:" doctor drug cure office". Translated query to English: "médecin mèdicament guérir cabinet".

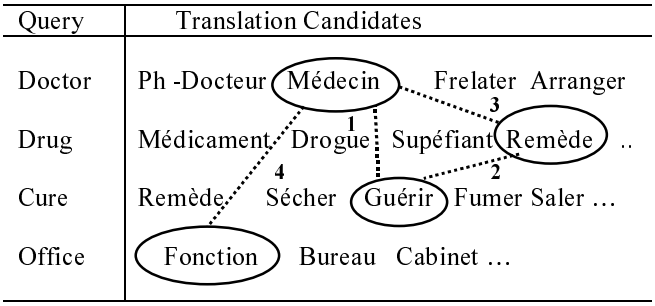


Figure 4: N-Term Disambiguation *(Second Method)*: Ranking and Disambiguation of Target Translations

Highest co-occurrence tendencies related to target translation candidates are as follows: (médecin, guérir), (guérir, remède), (remède, médecin) (médecin, fonction)…
Source French query: "doctor drug cure office". Translated query to English: "médecin remède guérir fonction".

translated queries (after translation) and the organization of source queries (before translation), using the co-occurrence tendency and the following estimations: Log-Likelihood Ratio (LLR) and Mutual Information (MI). LLR was used for *Bi_DIS*, disambiguation of consecutive pairs of source terms, without any ranking or selection (Sadat, 2001), for *LLR_DIS.bef*, the result of the first proposed disambiguation method (ranking source query terms, translation and disambiguation of target translations) and *LLR_DIS.aft*, the result of the second proposed disambiguation method (ranking and selecting target translation). In addition, MI estimation was applied to *MI_DIS.bef* and *MI_DIS.aft*, for the first and second proposed disambiguation methods. Query expansion was completed by the following methods: *Feed.bef_LLR*, which represents the result of adding a number of terms to the original queries and then performing a translation and disambiguation via *LLR_DIS.bef. Feed.aft*, is the result of query translation, disambiguation via *LLR_DIS.bef* method and then expansion. Finally, *Feed.bef_aft*, is the result of combined query expansion both before and after the translation and disambiguation via *LLR_DIS.bef.* In addition, we tested a query expansion before and after the disambiguation method MI_DIS.bef, together with the following methods: Feed.bef_MI, Feed.aft_MI and Feed.bef_aft_MI. Results and performance of these methods are described in Table 1. Figures 5 and 6 show the query translation/disambiguation using LLR and MI. Figures 7 and 8 show the query expansion for different combinations and estimations for the co-occurrence tendency (LLR or MI).

5.3. Discussion

The first column of Table 1 indicates the method. The second column indicates the number of retrieved relevant documents, and the third column indicates the precision averaged at point 0.10 on the Recall/Precision curve. The fourth column is the average precision, which is used as a basis for the evaluation. The fifth column is the R-precision and the sixth column represents the difference in term of average precision of the monolingual counterpart. Compared to the retrieval using original queries (English or French), *All_Tr* and *No_DIS* showed no improvement in term of precision, recall or average precision, whereas the simple two-term disambiguation *Bi_DIS* (disambiguation of consecutive pairs of source query terms) has increased the recall, precision and average precision by +1.71% compared to the simple dictionary translation without any disambiguation. On the other hand, the first proposed disambiguation method (ranking and selecting target translations) *LLR_DIS.aft,* showed a potential precision enhancement, 0.5012 at 0.10 and 90.82% average precision; however, recall was not improved (4131 relevant documents retrieved). The best performance for the disambiguation process was achieved by the second proposed disambiguation method (ranking source query terms and selecting target translations) *LLR_DIS.bef*, in average precision, precision and recall. The average precision was 101.51% of the monolingual counterpart, precision was 0.5144 at 0.10 and 436 relevant documents were retrieved. This suggests that ranking and selecting pairs for source query terms, is very helpful in the disambiguation process to select best target translations, especially for long queries of at least three terms. Results based on mutual information were less efficient compared to those using log-likelihood ratio. However, ranking source query terms before the translation and disambiguation resulted in an improvement in average precision, 100.91% of the monolingual counterpart. Although, query expansion before translation via *Feed.bef_LLR*/*Feed.bef_MI*, gave an improvement in average precision compared to the non-disambiguation method *No_DIS*, a slight drop in precision (0.4507/0.4394) and recall (413/405 relevant retrieved documents) was observed compared to *LLR_DIS.bef* or *MI_DIS.bef.* However, *Feed.aft_LLR/Feed.aft_MI* showed an improvement in average precision, 101.33%/101.25% compared to the monolingual counterpart and improved the precision (0.5153/0.5133 at 0.10) and the recall (433/ 430 retrieved relevant documents). Combined feedbacks both before and after translation yielded the best result, with an improvement in precision (0.5242 at 0.10), recall (434 retrieved relevant documents) and average precision, 102.89% of the monolingual counterpart when using LLR estimation. A disambiguation using MI for co-occurrence tendency yielded a good result, 103.53% of the monolingual counterpart for average precision. These results suggest that combined query expansion both before and after the proposed translation/disambiguation method improves the effectiveness of an information retrieval, when using a co-occurrence tendency based on MI or LLR.

	Rel Docs	at 0.10	A. Prec	R. Prec	% Mono
Mono_Fr (origin)	434	0.4178	0.2629	0.2925	100
Mono_En g (origin)	**433**	**0.4437**	**0.262**	**0.2663**	**100**
All_Tr	406	0.4285	0.2160	0.2573	82.19
No_DIS	429	0.4129	0.2214	0.2431	84.24
Bi_DIS	418	0.4115	0.2259	0.2769	85.95
LLR_DIS. Aft	431	0.5012	0.2387	0.2813	90.82
LLR_DIS. Bef	**434**	**0.5144**	**0.2679**	**0.3118**	**101.94**
MI_DIS.A ft	414	0.4507	0.2325	0.2556	88.47
MI_DIS.B ef	**429**	**0.5125**	**0.2652**	**0.3116**	**100.91**
Feed.bef_ LLR	413	0.4507	0.2309	0.2593	87.86
Feed.aft_ LLR	433	0.5153	0.2663	0.3165	101.33
Feed.bef_ aft_LLR	**436**	**0.5242**	**0.2704**	**0.3201**	**102.89**
Feed.bef_ MI	405	0.4394	0.2264	0.2521	86.14
Feed.aft MI	430	0.5133	0.2661	0.3074	101.25
Feed.bef aft_MI	**430**	**0.5160**	**0.2721**	**0.3077**	**103.53**

Table 1: Evaluations of the Translation, Disambiguation and Expansion Methods *(Different combinations with LLR and MI co-occurrence frequencies)*

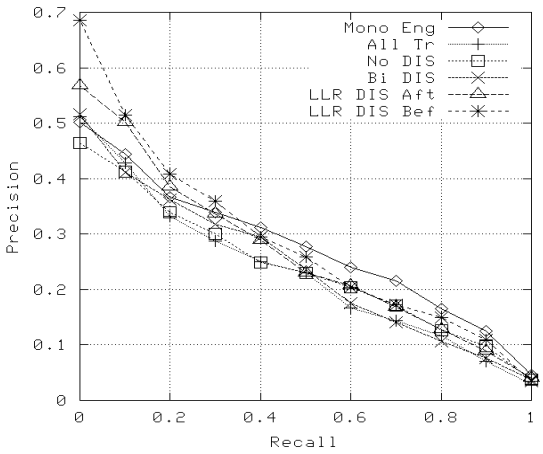


Figure 5: Recall/Precision Curves for the Query Translation/Disambiguation using LLR estimation

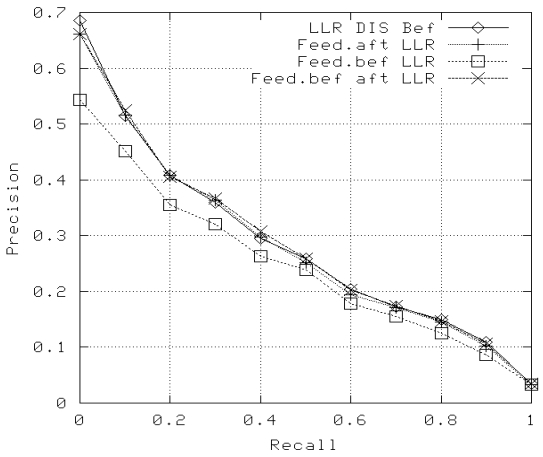


Figure 7: Recall/Precision Curves for the Query Expansion before and after the Translation/ Disambiguation using LLR estimation

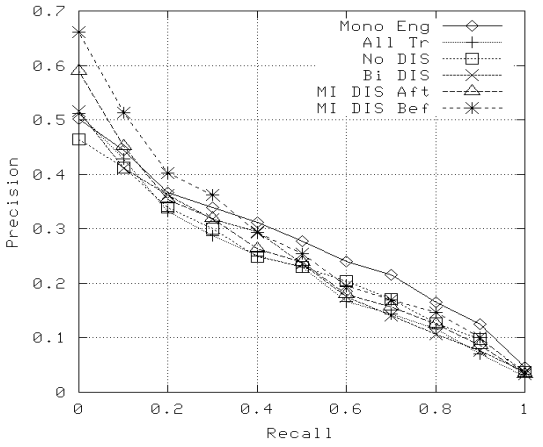


Figure 6: Recall/Precision Curves for the Query Translation/Disambiguation using MI estimation

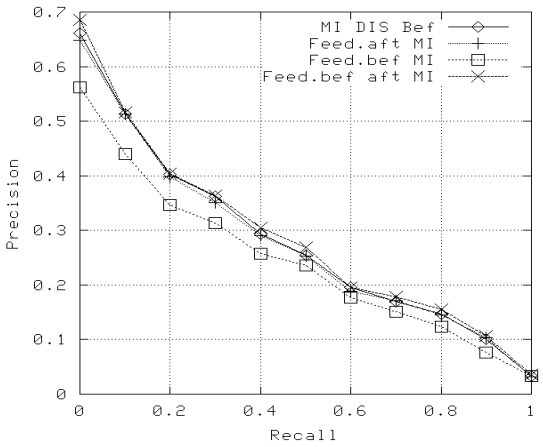


Figure 8: Recall/Precision Curves for the Query Expansion before and after the Translation/ Disambiguation using MI estimation

Thus, techniques of primary importance to this successful method can be summarized as follows:

- A statistical disambiguation method based on the co-occurrence tendency was applied first prior to translation, in order to eliminate misleading pairs of terms for translation and disambiguation. Then after translation, the statistical disambiguation method was applied in order to avoid incorrect sense disambiguation and to select best target translations.
- Ranking and careful selection are fundamental to the success of the query translation, when using statistical disambiguation methods.
- A combined statistical disambiguation method before and after translation provides a valuable resource for query translation and thus information retrieval,
- Log-Likelihood Ratio was found to be more efficient for query disambiguation than Mutual Information,
- A co-occurrence frequency to select an expansion term was evaluated using all terms of the original query, rather than using just one query term.
- Each type of query expansion has different characteristics and therefore combining various types of query expansion could provide a valuable resource for use in query expansion. This technique offered the greatest performance in average precision.
- These results showed that CLIR could outperform the monolingual retrieval. The intuition of combining different methods for query disambiguation and expansion, before and after translation, has confirmed that monolingual performance is not necessarily the upper bound for CLIR performance (Gao et al., 2001). One reason is that those methods have completed each other and that the proposed query disambiguation had a positive effect during the translation and thus retrieval. Combination to query expansion had an effect on the translation as well, because related words could be added.

The proposed combined disambiguation method prior to and after translation, was based on a selection of one target translation in order to retrieve documents. Setting a threshold in order to select more than one target translation is possible using weighting scheme for the selected target translations in order to eliminate misleading terms and construct an optimal query to retrieve documents.

6. Word Sense Disambiguation (WSD)

Word sense ambiguity is a pervasive characteristic of natural language and information retrieval. It is considered as one of the major causes of poor performance in Information Retrieval systems. We believe that a relationship between disambiguation, word sense ambiguity and IR, exists (Sanderson, 1994). Our proposed disambiguation method makes use of statistics data based on co-occurrence between words, which can be extracted from large language corpora. The motivation for this type of approach is the assumption that the used data will provide enough information to resolve most of word sense ambiguities encountered in practical applications. The acquisition of statistical data relies on the availability of training corpora, which is easier to acquire than parallel or aligned corpora. This approach could be well incorporated into Word Sense Disambiguation (WSD) when using dictionary-based translation. Moreover, it is easy to implement and cost effective. We believe that resolving word senses is worthwhile and could have a great impact on the recall and precision, especially, when training corpora are related to particular or different subject areas (Krovetz and Croft, 1992).

7. Conclusion

Dictionary-based method is attractive for several reasons. This method is cost effective and easy to perform, resources are readily available and performance is similar to that of other Cross-Language Information Retrieval methods. Ambiguity arising from failure to translate queries is largely responsible for large drops in effectiveness below monolingual performance (Ballesteros and Croft, 1997). The proposed disambiguation approach of using statistical information from language corpora to overcome limitation of simple word-by-word dictionary-based translation has proved its effectiveness, in the context of information retrieval. A co-occurrence tendency based on a log-likelihood ratio has showed to be more efficient than the one based on mutual information. The combination of query expansion techniques, both before and after translation through relevance feedback improves the effectiveness of simple word-by-word dictionary translation. We believe that the proposed disambiguation and expansion methods will be useful for simple and efficient retrieval of information across languages.

Ongoing research includes a search for additional methods that may be used to improve the effectiveness of information retrieval. Such methods may include the combination of different resources and techniques for optimal query expansion across languages. In addition, thesauri and relevance feedbacks will be studied in greater depth. A good word sense disambiguation model will incorporate several types of data source that complete each other, such as a part-of-speech tagger into statistical models. Finally, an approach to learning from documents categorization and classification in order to extract relevant expansion terms will be examined in the future.

Acknowledgments

The present study is supported in part by the Ministry of Education, Culture, Sports, Science and Technology of Japan, under grants 11480088, 12680417 and 12208032, and by the CREST program of the JST Corporation (Japan Science and Technology). We would like to thank Dr Claude de Loupy and all reviewers for their helpful comments on the earlier version of this paper.

References

Ballesteros, L. and Croft, W. B. 1998. Resolving Ambiguity for Cross-Language Retrieval. In proceedings of the 21st ACM SIGIR Conference.P:64-71.

Church, K. W. and Hanks, P. 1990. Word association Norms, Mutual Information and Lexicography. Computational Linguistics, Vol 16 No1. P: 22-29.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. Computational linguistics, Vol.19.,No.1. P: 61-74.

Gale, W. A. and Church, K. 1991. Identifying word correspondences in parallel texts. In proceedings of the 4th DARPA Speech and Natural Language Workshop. P: 152-157.

Gao, J., Nie,J.Y., Xun, E., Zhang, J., Zhou, M., Huang, C. 2001. Improving query translation for Cross-Language Information Retrieval using statistical models. In proceedings of the 24st ACM SIGIR Conference. P: 96-104.

Hull, D. and Grefenstette, G. 1996. Querying across languages. A dictionary-based approach to Multilingual Information Retrieval. In proceedings of the 19th ACM SIGIR Conference. P:49-57.

Hull, D. 1998. A weighted boolean model for Cross-Language text Retrieval. In G. Grefenstette editor: Cross-Language Information Retrieval, chapter 10. Kluwer Academic Publishers.

Hutchins, J. and Sommers, J. 1992. Introduction to Machine Translation. Academic Press.

Krovetz, R. and Croft, W. 1992. Lexical ambiguity and information retrieval. ACM Transactions on Information Systems, 10 (2). P: 115-141.

Loupy, C., Bellot, P., El-Beze, M. and Marteau, P.-F. 1998. Query expansion and classification of retrieved documents. In Proceedings of TREC-7. NIST Special Publication.

Maeda, A., Sadat, F., Yoshikawa, M. and Uemura, S. 2000. Query term disambiguation for Web Cross-Language Information Retrieval using a search engine. In Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages. P: 25-32.

Oard, D.W. 1997. Alternative approaches for Cross-Language Information Retrieval. In Working notes of the AAAI Symposium on Cross-Language Text and Speech Retrieval. Stanford University, USA. http://www.glue.umd.edu/~oard/research.html

Sadat, F., Maeda, A., Yoshikawa, M. and Uemura, S. 2001. Query expansion techniques for the CLEF bilingual track. In Working Notes for the CLEF 2001 Workshop. P: 99-104.

Sanderson, M. 1994. Word Sense Disambiguation and Information Retrieval. ACM Special Interest Group on Information Retrieval. P: 142-151.

Yamabana, K., Muraki, K., Doi, S. and Kamei, S. 1996. A language conversion Front-End for Cross-Linguistic Information Retrieval. In Proceedings of SIGIR Workshop on CLIR, Zurich, Switzerland. P: 34-39.

Vossen, P. EuroWordNet. 1998. A Multilingual database with lexical semantic networks. Kluwer Academic Publishers.

Semantic enrichment for information extraction using word sense disambiguation

Bernard Jacquemin, Caroline Brun and Claude Roux

Xerox Research Centre Europe
6, chemin de Maupertuis, 38 240 Meylan, France
{Bernard.Jacquemin,Caroline.Brun,Claude.Roux}@xrce.xerox.com

Abstract

External linguistic resources have been used for a very long time in information extraction. These methods enrich a document with data that are semantically equivalent, in order to improve recall. For instance, some of these methods use synonym dictionaries. These dictionaries enrich a sentence with words that have a similar meaning. However, these methods present some serious drawbacks, since words are usually synonyms only in restricted contexts. The method we propose here consists of using word sense disambiguation rules (WSD) to restrict the selection of synonyms to only these that match a specific syntactico-semantic context. We show how WSD rules are built and how information extraction techniques can benefit from the application of these rules.

1. Introduction

In today's world, the society of communications is gaining in importance every day. The amount of electronic documents – mainly by Internet, but not only – grows more and more. With this increase, no one is able to read, classify and structure those documents so that the requested information can be reached when it is needed. Therefore we need tools that reach a shallow understanding of the content of these texts to help us to select the requested data.

The process of understanding a document consists in identifying the concepts of the document that correspond to requested information. This operation can be performed with linguistic methods that permit the extraction of various components related to the data that are requested.

Since the beginning of the '90s, several research projects in information extraction from electronic text have been using linguistic tools and resources to identify relevant elements for a request. The first ones, based on domain-specific extraction patterns, use hand-crafted pattern dictionaries (CIRCUS (Lehnert, 1990)). But systems were quickly designed to build extraction pattern dictionaries automatically. Among these systems, AutoSlog (Riloff, 1993; Riloff and Lorenzen, 1999) builds extraction pattern dictionaries for CIRCUS. CRYSTAL (Soderland et al., 1995) creates extraction patterns lists for BADGER, the successor of CIRCUS. These learners use hand-tagged specific corpora to identify structures containing the relevant information. The syntactic structure used by CRYSTAL is more subtle than the one used by AutoSlog. CRYSTAL is able to make the most of semantic classes. WHISK (Soderland, 1999) is one of the most recent information extraction system. WHISK has been designed to learn which data to extract from structured, semi-structured and free text [1]. A parser and a semantic tagger have been implemented for free text. This system is the only one to process all of these three categories of text.

These methodologies need domain-specific pattern dictionaries that must be built for each different kind of information. However, none of these methods can be directly applied to generic information. Thus we decide to bypass these two obstacles: our approach is based on the utilization of an existing electronic dictionary, in order to expand the data in a document to equivalent forms extracted from that dictionary.

Our method deals with the identification of semantic contents in documents through a lexical, syntactic and semantic analysis. It then becomes possible to enrich words and multi-word expressions in a document with synonyms, synonymous expressions, semantic information etc. extracted from the dictionary.

2. Problems and Prospects

As for a lot of methodologies developed for natural language processing, the results of a method of information extraction are evaluated by two measures: precision and recall. Precision is the ratio of correctly extracted items to the number of items both correctly and erroneously extracted from the text; noise is the ratio of the faulty extracted items to all the achieved extractions. Recall is the ratio of correctly extracted items to the number of items actually present in the text. The problem consists in improving both precision and recall.

2.1. Recall improvement

A usual technique to improve the recall consists of enriching a text with a list of synonyms or near-synonyms for each word of that text. For example, all the synonyms of "climb" would be added to the document, even though some of those meanings have a remote semantic connection to the text. By this kind of enrichment, all the ways to express the same token (but not the same meaning) are taken into account.

This type of enrichment can be extended to synonymous expressions with a robust parser: syntactic dependencies and their arguments (the tokens belonging to the selected expression) are enlarged to dependencies that are generated out of the corresponding synonymous expressions.

[1] We use the term "structured text" to refer to what the database community calls semi-structured text; "semi-structured text" is ungrammatical and often telegraphic text that does not follow any rigid format; "free text" is simply grammatical text (Soderland, 1999).

The recall is usually optimised to the detriment of the precision with those techniques, since most words within a set of synonyms are themselves polysemous and are seldom equivalent for each of their meanings. Thus, a simply adding of all those polysemous synonyms in a document introduces meaning inconsistencies. Noise may stem from these inconsistencies.

2.2. Reduction of noise – Precision improvement

We notice that improving the recall using synonyms may often increase the noise. Although identified in the domain of IE, this problem is not yet solved and it has a negative influence on the system effectiveness. Our purpose is to use the linguistic context of the polysemous tokens to identify their meanings and select contextual synonyms or synonymous expressions. This approach should improve the precision in comparison with adding all the synonyms.

Sentences in the text:

La température **grimpe**.
(The temperature is climbing.)

Corresponding set of synonyms:

escalader	*monter*
(to climb)	*(to go up)*
sauter	*augmenter*
(to jump)	*(to increase)*
se hisser sur	
(to heave oneself up onto)	

Sentences resulting from the enrichment:

La temperature escalade.
La temperature ***monte****.*
La temperature saute.
La temperature ***augmente****.*
La temperature se hisse sur (???).

Figure 1: Enrichment by a list of synonyms.

For example, the dictionary [2] entry for the word *grimper* contains a set of 5 synonyms. If we use these synonyms to enrich the original text, we obtain five variations of the original sentence. Only the second and the fourth of the enriching variations are accurate in this context. The meteorological context associated with the word *température* in the dictionary should correctly discriminate the synonyms in this context: in the dictionary, each synonym of a lemma is associated with a meaning of this lemma and with the typical linguistic context of the lemma in this sense.

Consequently, we decided to use the linguistic context of the words that can be enriched to discriminate which synonyms should be used and which should not. The synonyms are stored in the dictionary according to the sense of each lemma. So, the task amounts to performing a lexical semantic disambiguation of the text and using synonymous expressions in the selected meanings to enrich the document.

3. Enrichment method by WSD

3.1. Our experience in WSD

We previously have developed a range of tools and techniques to perform Word Sense Disambiguation (WSD), for French and English. The basic idea is to use a dictionary as a tagged corpus in order to extract semantic disambiguation rules, (Brun et al., 2002; Brun, 2000; Brun and Segond, 2001; Dini et al., 1998; Dini et al., 2000). Since electronic dictionaries exist for many languages and they encode fine-grained reliable sense distinctions, be they monolingual or bilingual, we decided to take advantage of this detailed information in order to extract a semantic disambiguation rule database [3]. The disambiguation rules associate each word with a sense number taking the context into account. For bilingual dictionaries the sense number is associated with a translation, for monolingual dictionaries with a definition. WSD is therefore performed according to sense distinctions of a given dictionary. The linguistic rules have been created using functional dependencies provided by an incremental shallow parser (IFSP, (Ait-Mokhtar and Chanod, 1997)), semantic tags from an ontology (45 classes from WordNet (Feldbaum, 1998) for English) as well as information encoded in SGML tags of dictionaries. This method comprises two stages, rule extraction and rule application.

- Rule extraction process: for each entry of the dictionary, and then for each sense of the entry, examples are parsed with the IFSP shallow parser. The shallow parsing task includes tokenization, morphological analysis, tagging, chunking, extraction of functional dependencies, such as subject and object (SUBJ(X, Y), DOBJ (X, Y)), etc. For instance, parsing the dictionary example attached to one particular sense $\mathbf{S}_i$ of *drift* :

  1)The country is drifting towards recession.

  Gives as output the following chunks and dependencies :

  [SC [NP The country NP]/SUBJ :v is drifting SC] [PP towards recession PP] SUBJ(country, drift) VMOD-OBJ(drift, towards, recession)

  Using both the output of the shallow parser and the sense numbering from the dictionary we extract the following semantic disambiguation rule: When the ambiguous word "drift" has *country* as subject and/or *toward recession* as modifier, it can be disambiguated with its sense $\mathbf{S}_i$. We repeat this process as all dictionary example phrases in order to extract the word level rules, so called because they match the lexical context.

[2] The dictionary we use is a French electronic one (Dubois and Dubois-Charlier, 1997). We will give a more detailed information about it later.

[3] The English dictionary contained 39 755 entries and 74 858 senses, ie a polysemy of 1.88; the French dictionary contained 38 944 entries and 69 432 senses, ie a polysemy of 1.78

Finally, for each rule already built, we use semantic classes from an ontology in order to generalize the scope of the rules. In the above example the subject "country" is replaced in the semantic disambiguation rule by its ambiguity class. We call ambiguity class of a word, the set of WordNet tags associated with it. Each word level rule generates an associated class level rule, so called because it matches the semantic context: when the ambiguous word "drift" has a word belonging to the WordNet ambiguity class *noun.location* and *noun.group* as subject and/or a word belonging to the WordNet ambiguity class *noun.shape*, *noun.act*, and *noun.state* as modifier, it disambiguates with its sense $\mathbf{S}_i$. Once all entries are processed, we can use the disambiguation rule database to disambiguates new unseen texts. For French, semantic classes (69 distinctive characteristics) provided by the *AlethDic* dictionary (Gsi-Erli, 1994) have been used with the same methodology.

- Rule application process: The rule applier matches rules of the semantic database against new unseen input text using a preference strategy in order to disambiguate words on the fly. Suppose we want to disambiguate the word drift, in the sentence:

  2) In November 1938, after Kristallnacht, the world drifted towards military conflict.

  The dependencies extracted by the shallow parser, which might lead to a disambiguation, i.e., which involve *drift*, are:

  SUBJ(world, drift)
  VMODOBJ(drift, towards, conflict)

  The next step tries to match these dependencies with one or more rules in the semantic disambiguation database. First, the system tries to match lexical rules, which are more precise. If there is no match, then the system tries the semantic rules, using a distance calculus between rules and semantic context of the word in the text [4]. In this particular case, the two rules previously extracted match the semantic context of *drift*, because *world* and *country* shares semantic classes according to WordNet, as well as *conflict* and *recession*.

The methodology attempts to avoid the data acquisition bottleneck observed in WSD techniques. Thanks to this methodology, we built all-words (within the limits of the used dictionary) unsupervised Word Sense Disambiguator for French (precision: 65%, recall: 35%) and English (precision: 79%, recall: 34%).

3.2. Xerox Incremental Parser (XIP)

IFSP, which was used in the first experiments on semantic disambiguation at Xerox, has been implemented with transducers. Transducers proved to be an interesting formalism to implement quickly an efficient dependency parser, as long as syntactic rules would only be based on POS. The difficulty of using more refined information, such as syntactic features, drove us to implement a specific platform that would keep the same strategies of parsing as in IFSP, but would no longer rely on transducers.

This new platform (Ait-Mokhtar et al., 2001; Roux, 1999) comprises different sorts of rules that chunk and extract dependencies from a sequence of linguistics tokens, which is usually but not necessarily a sentence. The grammar of French that has been developed computes a large number of dependencies such as *Subject, Object, Oblique, NN* etc. These dependencies are used in specific rules, the disambiguation rules, to detect the syntactic and semantic information surrounding a given word in order to yield a list of words that are synonyms according to that context. Thus, a disambiguation rule manipulates together a list of semantic features originating from dictionaries, and a list of dependencies that have been computed so far. The result is a list of contextual synonyms.

If (Dependency$_0$(t, t^0) & ...& Dependency$_n$(t,t^k) & ... attribute$_p$(t^j)=v^u)

synonym(t) = s^0,...,s^n.

where

t^0,...,t^n is a list of token

s^0,...,s^n a list of synonyms.

Example:

- La température grimpe.
  (the temperature is climbing)
- La température augmente.
  (the temperature is rising)
- L'alpiniste grimpe le mont Ventoux.
  (the alpinist climbs the mount Ventoux)
- ???L'alpiniste augmente le mont Ventoux.
  (???the alpinist raises the mount Ventoux)

Figure 2: Application of a disambiguation rule for enrichment.

The contextual synonymy between *grimper* and *augmenter* can be defined with the following rule. The feature *MTO* is one of the semantic features that are associated with the entries of the Dubois dictionary. This feature is associated with each word that is connected to meteorology, such as *chaleur, froid, température* (heat, cold, température).

if (Subject(*grimper*, X) AND feature(X, *domain*)=MTO) synonym(*grimper*) = *augmenter*.

This rule applies on the above first example, *La température grimpe*, but fails to apply on the third sentence, *L'alpiniste grimpe le mont Ventoux*, since the subject does not bear the MTO feature.

[4] The first parameter of this metric is the intersection of the rule classes and the context classes; the second one is the union of the rule classes and the context classes. Distance equals the ratio of intersection to union.

3.3. Which WSD for which enrichment?

3.3.1. A very rich dictionary information

The new robust parser offers a flexible formalism and the possibility to handle semantic or other features. In addition to this parser, the semantic disambiguation now uses a monolingual French dictionary (Dubois and Dubois-Charlier, 1997). This dictionary contains many kind of information in the lexical field as well as in the syntactic or the semantic one. From the 115 229 entries of this dictionary, we can only use the 38 965 ones that are covered by the morphological analyser. These entries represent 68 588 senses, ie a polysemy of 1.76.

We build lexico-syntactic WSD rules using the methodology presented above (cf. section 3.1.): examples of the dictionary are parsed; extracted syntactic relations and their arguments are used to create the rules. We also make the most of the domain indication (171 different domains) to generalize the example rules (see later for details) – as previously done using WordNet for the English WSD and by AlethDic for the French one (Brun et al., 2002).

We use the specificity of the dictionary to improve the disambiguation task as far as possible in order to maximize the enrichment of the documents. The information of this dictionary is divided into several fields: domain, example, morphological variations, derived or root words, synonyms, POS, meaning, estimate of use frequency in the common language; in the verbal part of the dictionary only, syntactico-semantic class and subcategorization patterns of the arguments of the verb. Resulting WSD rules are spread over three levels reflecting the abstraction register of the dictionary fields.

3.3.2. Disambiguation rules at various levels

We build a disambiguation rule database at three levels: rules at word level (23 986), rules at domain level (22 790) and rules at syntactico-semantic level (40 736).

Word level rules use lexical information from the examples. They correspond to the basic rules in the previous system, which use constraints on words and syntactic relations. These dependencies are extracted from the illustrative examples from the dictionary.

L'avion de la société **décrit** un large cercle avant de (…)
*(The company's plane **describes** a wide circle before (…))*
SUBJECT(décrire,avion)
OBJECT(décrire,cercle)

Example in the dictionary for the entry "décrire":
L'avion décrit un cercle.
(The plane describes a circle.)
SUBJECT(décrire,avion)
OBJECT(décrire,cercle)

Figure 3: WSD at word level.

Rules at domain level are generalized from word level rules: instead of using the words of the examples as arguments of the syntactic relations in the rules, we replace them by the domains they belong to. These rules correspond to the class level rules in the previous system, but an improvement in comparison with them is that in some cases, we can discriminate the right domain if the argument is polysemous. This is mainly due to the internal consistency of the dictionary that enables the correspondences of domain across different arguments of a dependency. The consistency should help to reduce the noise.

L'escadrille décrit son approche vers l'aéroport où (…)
(The squadron describes its approach to the airport where (…))
SUBJECT(décrire,escadrille[dom:AER])
OBJECT(décrire,approche[dom:LOC])

Example in the dictionary for the entry "décrire":
L'avion décrit un cercle.
(The plane describes a circle.)
SUBJECT(décrire,avion[dom:AER])
OBJECT(décrire,cercle[dom:LOC])

Figure 4: WSD at domain level.

We don't rule out the possibility of using other lexico-semantic resources to generalize or expand this kind of rules, as we did previously using French EuroWordNet or AlethDic. These lexicons present the advantage of a hierarchical structure that doesn't exist for the domain field in the Dubois dictionary. Nevertheless, we will encounter the problem of the mapping of the various resources used by the system to avoid inconsistencies between them, as shown in (Ide and Véronis, 1990; **?**; Brun et al., 2002).

The third level of the rules currently in use in the semantic disambiguator is the syntactico-semantic one. The abstraction level of these rules is even higher than in the domain level. They are built from a syntactic pattern of subcategorization that indicates the typical syntactic construction of the current entry in its current meaning. Although the distinction between the arguments is very general – they are differentiated from human, animal and inanimate – our examination of the verbal dictionary indicates that, for 30% of the polysemous entries, this kind of rules is sufficient to choose the appropriate meaning.

3.4. Enrichment at various levels

WSD is not an end in itself. In our system, it is a means to select appropriate information in the dictionary to enrich a document. The quality and the variety of this enrichment vary according to the quality and the richness of the information in the dictionary. The variety of information allows several kind of enrichment.

For the specific task of information extraction, an index of the documents whose information is likely to be extracted is built. It allows the classification of all the linguistic realities extracted from text analysis. These realities are listed according to the XIP-formalism: syntactic relations,

L'escadrille décrit son approche vers l'aéroport where (...)
(The squadron describes its approach to the airport where(...))
SUBJECT(décrire,escadrille[dom:AER])
OBJECT(décrire,approche[dom:LOC])

Subcategorisation for the entry "décrire":
Transitive verb;
Subject inanimate.
SUBJECT(décrire,?[subcat:inanimate]) & OBJECT(décrire,?)

Figure 5: WSD at lexico-semantic level.

arguments, and features attached to the arguments. The enrichment is done inside the index because dependencies can be added without affecting the original document.

3.4.1. Lexical level

Replacing a word by its contextual synonyms is the easiest way to perform enrichment. This method of recall improvement is very common in IE, but in our system, the enrichment is targeted according to the context thanks to the semantic disambiguation. This process often reduces the noise. The enrichment is achieved by copying the dependencies containing the disambiguated word and by replacing this word by one of its synonyms.

La température grimpe.
(The temperature is climbing.)

Original index:
SUBJECT(grimper,température)

Set of targeted synonyms:
monter, augmenter.

Enriched index:
SUBJECT(grimper,température)
SUBJECT(monter,température)
SUBJECT(augmenter,température)

Figure 6: Enrichment at lexical level.

3.4.2. Lexico-syntactic level

The lexico-syntactic level of enrichment is more complex to achieve. The task consists in replacing a word by a multi-word expression (more than 14 000 synonyms are multi-word expressions in our dictionary) or in replacing a multi-word expression by a word, taking into account the words (lexical) and the dependencies between them (syntactic):

- Replacing a word by a multi-word expression (see figure 7):
  - Parse the multi-word expression to obtain dependencies;
  - Match the corresponding dependencies in the text;
  - Instantiate the missing arguments with the text arguments.
- Replacing a multi-word expression by a word:
  - Identify the POS of the word;
  - Select dependencies implying one and only one word of the multi-word expression;
  - Eliminate dependency where this word has a different POS;
  - Replace this word with its synonym in the remaining dependencies.

Le spécialiste a édité un manuscrit très abîmé.
(The specialist published a very damaged manuscript.)

Original index:
SUBJECT(éditer,spécialiste)
OBJECT(éditer,manuscrit)

Targeted synonymous expression:
établir l'édition critique de

Extracted dependencies from the expression:
SUBJECT(établir,?)
OBJECT(établir,édition)
EPITHET(édition,critique)
PP(édition,de,?)

Enriched index:
SUBJECT(éditer,spécialiste)
OBJECT(éditer,manuscrit)
SUBJECT(établir,spécialiste)
OBJECT(établir,édition)
EPITHET(édition,critique)
PP(édition,de,manuscrit)

Figure 7: Enrichment at lexico-syntactic level.

Since our work is based on the Dubois dictionary – whose entries are single words – most of the enrichment is one-to-one word. When a multi-word expression appears in the synonyms list, a single word has to be replaced by a multi-word expression, and the inverse process can be achieved if necessary. The complex case of replacing a multi-word expression by another multi-word expression could arise, but we never encounter this situation. The replacement of a multi-word expression by another is not yet implemented because of the complexity of the process. Nevertheless, the system relies on relations and arguments that are easy to handle, very simple and modular. These characteristics should allow us to bypass the inherent complexity of these structures.

3.4.3. A semantic level example

Syntactico-semantic fields in the dictionary allow a third enrichment level. The syntactico-semantic class structure contains very useful information that makes it possible to link verbs that are semantically related but lexically and syntactically very different. It might be interesting to semantically link *vendre* ("to sell", class D2a) and *acheter* ("to buy", class D2c) even though their respective actors are inverted. For example, *le marchand vend un produit au client* (the trader sells a product to the customer) bears the same meaning as *le client achète un produit au marchand* (the customer buys a product from the trader). The semantic class gives a general meaning of the verb(D2, meaning *donner, obtenir*, to give, to obtain), while the syntactic pattern (a for *vendre*: *fournir qc qn*, to supply so with sth, transitive with a oblique compliment, c for *acheter*: *prendre qc qn*, to take sth to so, transitive with a oblique compliment) yields the semantic realization.

Le papa offre un cadeau à sa fille.
(The father is giving a present to his daughter.)

Original index:
SUBJECT(offrir,papa)
OBJECT(offrir,cadeau)
OBLIQUE(offrir,fille)

offrir 01: D2a (to give sth to sb)
D2a corresponds to D2e (receive, obtain sth from sb).
recevoir 01: D2e

Enriched index:
SUBJECT(offrir,papa)
OBJECT(offrir,cadeau)
OBLIQUE(offrir,fille)
SUBJECT(recevoir,fille)
OBJECT(recevoir,cadeau)
????(recevoir,de,papa)

Figure 8: Enrichment at semantic level.

In a same perspective, a syntactico-semantic class constitutes another synonym set. Since this set is too general and too imprecise, it cannot be used to enrich a document. Still, it can be used as a last resort to enrich the query side when other methods have failed. We will not use this set as enrichment, but only to match a query by the class if the enrichment fails.

4. Evaluation

Though the method presented in this article is based on previous works, the use of other tools and lexical resource may have extended the potential of WSD rules. In particular, it is possible that the number of domains increase precision, and the use of subcategorization patterns may ensure more general rules to increase recall.

The partial evaluation we performed concerns 604 disambiguations in a corpus of 82 sentences from the French newspaper *Le Monde*. Precision in WSD is ratio of correct disambiguations to all disambiguations performed; recall is ratio of correct disambiguations to all possible disambiguations in the corpus. We distinguish the mistakes due to the method and the ones linked to our analysis tools in order to identify what we have to improve in order to increase the performance. These results are promising since both precision and recall are better than in the previous system.

Tokenization mistakes	44	7.28%
Tagging mistakes	19	3.15%
Parsing mistakes	9	1.49%
WSD mistakes	84	13.91%
Precision	**448**	**74.17%**
Recall		**43.61**%

Table 1: WSD method evaluation.

We note some remarks about this evaluation:

1. The lexicon used to perform tokenization has been modified in order to include additional information from the dictionary. We noticed during this evaluation some problems of coverage;

2. For this first prototype, we do not yet establish a strategy for cases in which multiple rules match. If more than one rule can be applied to the context, the sense is randomly chosen among the ones suggested by the matching rules [5];

3. Conversely, we do not yet try a strategy using the domain of disambiguated words as a general context to choose the corresponding meaning of a word to disambigate.

During the evaluation, we also notice that when a result was correct, the suggested synonymous expressions were always correct for the disambiguated word in this context. Our method for an optimized enrichment is validated.

5. Conclusion

In this paper, we present an original method for processing documents, preparing the text for information extraction. The goal of this processing is to expand each concept by the largest list of contextualy synonymous expressions in order to match a request corresponding to this concept.

Therefore, we implement an enrichment methodology applied to words and multi-word expressions. In order to perform the enrichment task, we have decided to use WSD to contextually identify the appropriate meaning of the expressions to expand. Inconsistent enrichment by synonyms is currently known as a major cause of noise in Information Extraction systems. Our strategy lets the system target the enriching synonymous expressions according to the semantic context. Moreover, this enrichment is achieved not

[5] This random choice is only performed for this evaluation and not in a IE perspective, since noise is better than silence in this field.

only with single synonymous words, but also with multi-word expressions that might be more complex than simple synonyms.

The WSD task and the resulting enrichment stage are achieved using syntactic dependencies extracted by a robust parser: the WSD is performed using lexico-semantic rules that indicate the preferred meaning according to the context. The linguistic information extracted from the analysis of the documents is indexed for the IE task. This index also stores additional new dependencies stemming from the enrichment process.

The utilization of a unique, all-purpose dictionary to achieve WSD and enrichment ensures the consistency of the methodology. Nevertheless, the information quality and richness of the dictionary might determine the system effectiveness.

The evaluation validates the quality of our method, which allows a great deal of lexical enrichment with less noise than is introduced by other enrichment methods. We have also indicated some ways our method could be expanded and our analysis tools could be improved. Our next step will be to test the effect of the enrichment in an IE task.

The method is designed to achieve a generic IE task, and the tools and resources are developed to process text data at a lexical level as well as at a syntactic or semantic level.

6. References

Salah Ait-Mokhtar and Jean-Pierre Chanod. 1997. Subject and object dependency extraction using finite-state transducers. In *Workshop on automatic Information Extraction and the Building of Lexical Semantic Resources, ACL*, pages 71–77, Madrid, Spain.

Salah Ait-Mokhtar, Jean-Pierre Chanod, and Claude Roux. 2001. A multi-input dual-entry point dependency parser. In *Proceedings of the International Workshop of Parsing Technology*, Beijing, China. IWPT-01.

Caroline Brun and Frédérique Segond. 2001. Semantic encoding of electronic documents. *International Journal of Corpus Linguistic*, 6:1:79–97.

Caroline Brun, Bernard Jacquemin, and Frédérique Segond. 2002. Exploitation de dictionnaires électroniques pour la désambiguisation sémantique lexicale. *TAL, special issue on Lexiques Sémantiques*, 42:3:to appear.

Caroline Brun. 2000. A client/server architecture for word sense disambiguation.

Luca Dini, Vittorio Di-Tomaso, and Frédérique Segond. 1998. Error driven word sense disambiguation. In *proceedings of COLING/ACL98*, pages 320–324, Montreal, Canada.

Luca Dini, Vittorio Di-Tomaso, and Frédérique Segond. 2000. Ginger II: an example-driven word sense disambiguato. *Computer and the Humanities, special issue on Senseval*, 34:121–129.

Jean Dubois and Françoise Dubois-Charlier. 1997. *Dictionnaire des verbes français*. Larousse, Paris. This dictionary exists in an electronic version and is accompanied by the corresponding electronic Dictionnaire des mots français.

Christiane Feldbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, (MA).

Gsi-Erli. 1994. *Le dictionnaire AlethDic*. Erli.

Nancy Ide and Jean Véronis. 1990. Mapping dictionaries: A spreading activation approach. In *Proceedings of the 6th Annual Conference of the Centre for the New Oxford English Dictionary*, pages 52–64, Waterloo, Ontario.

Wendy Lehnert. 1990. Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. In J. Barnden and J. Pollack, editors, *Advances in Connexionist and Natural Computation Theory*, volume 1, pages 135–164. Ablex Publishers, Norwood, NJ.

Ellen Riloff and Jeffrey Lorenzen. 1999. Extraction-based text categorization: generating domain-specific role relationships automatically. In T. Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer Academic Publisher.

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816. AAAI Press / MIT Press.

Claude Roux. 1999. Phrase-driven parser. In *Proceedings of VEXTAL'99*, Venezia, Italia. VEXTAL'99.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 811–816. IJCAI-95.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34:233–272.

Word Sense Disambiguation Using Semantic Sets based on WordNet

Ganesh Ramakrishnan

Computer Sc. & Engg.
Indian Institue of Technology
Mumbai - 400076

Pushpak Bhattacharyya

Computer Sc. & Engg.
Indian Institue of Technology
Mumbai - 400076

Abstract

This paper presents an automatic method for resolving the lexical ambiguity of nouns in any free-flowing text. The method exploits the noun taxonomy present in the WordNet and also the relative position of nouns in the given text, to construct semantic sets from the text. The semantic set has been defined as a collection of senses of words in given text that are related through the WordNet. Two different concepts of semantic distance between words have been explored and used for disambiguation. Hand-tagging of text and training are not required by the method presented in this paper. The method has been tested against SemCor, the tagged version of the Brown corpus and compared with previous unsupervised WSD algorithms. The method is supported by good empirical results.

1 Introduction

Any language uses words with multiple meanings. Before Information Retrieval or Semantic analysis of texts, it is essential to determine the true senses of those words. The problem of determining the right sense of words, in a context, is called *Word Sense Disambiguation* (WSD).

The typical approaches to the problem of WSD can be classified into 3 types: (1) *Supervised*, (2) *Unsupervised* and (3) *Cross-Lingual*.

Supervised Methods require resources like semantically annotated corpora to train the WSD system, and lexical resource like WordNet which provides the sense numbers using which the annotations are made. These algorithms, like the ones considered in [1], [2] and [3] use the corpora like Grolier's encyclopedia [1] or private sense-tagged data-sets [2]. However, the semantically annotated corpora are Laborious to construct and expensive, since tagging is done manually or at most semi-automatically.

Unsupervised Methods consider the statistically relevant co-occurance of individual keywords as classes and generate a class based model to predict which will be the most likely class to follow a particular keyword. The class is treated as an equivalent of sense. Unsupervised WSD methods can be further classified into two types, viz. WSD that makes use of the information provided by machine readable dictionaries: this is the case with the work reported by [10], [14], [4], [12] and [11]. And WSD that uses information gathered from raw corpora (unsupervised training methods); [1] and [13] presented unsupervised WSD methods using raw corpora.

From a multilingual point of view, word sense disambiguation is nothing more than determining the appropriate *translation* of a word or lexical item. Thus, translation presupposes word sense disambiguation. Word translation only requires only that the words should be expressing the same meaning. However, it is not necessary to know the exact meaning of the words. See [7] for further details.

2 WordNet

WordNet[9] is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. WordNet was developed by the Cognitive Science Laboratory at Princeton University.

The WordNet consists of synsets arranged in semantic relationships with one another, through *hypernymy, hyponymy, holonymy, meronymy, synonymy* and *antonymy* relationships. In our discussion, we use WordNet as the only lexical resource and all the *senses* are with respect to the WordNet.

3 Semantic Set

Below is a sample text of 100 words, from the *Brown Corpus*, with some nouns underlined.

In the WordNet sub-graph in figure 2, the relationship between these nouns is shown. The words marked in ellipses are words that actually occur in the text.

The Fulton_County_Grand_Jury said Friday an investigation of Atlanta 's recent primary_election produced no *evidence* that any irregularities took_place. The jury further said in *term* end presentments that the City_Executive_Committee which had over-all charge of the election deserves the praise and thanks of the City_of_Atlanta for the manner in which the election was conducted The September-October *term* jury had been charged by Fulton Superior_Court_Judge_Durwood_Pye to investigate reports of possible irregularities in the hard-fought primary which was won by Mayor-nominate_Ivan_Allen_Jr ..... It recommended that Fulton legislators act to have these laws studied and revised to the end of modernizing and improving them. The grand_jury commented on a number of other *topics* among them the Atlanta and Fulton_County purchasing_departments which it said are well operated and follow generally accepted practices which inure to the best interest of both governments However the jury said it believes these two offices should be combined to achieve greater efficiency and reduce the cost of administration ... Implementation of Georgia 's automobile title *law* was also recommended by the outgoing jury It urged that the next Legislature provide enabling funds and re-set the effective date so_that an orderly implementation of the *law* may be effected. ... This is one of the major items in the Fulton_County general assistance *program* the jury said but the State_Welfare_Department has seen_fit to distribute these funds through the welfare departments of all the counties in the state with the exception of Fulton_County which receives none of_this money The jurors said they realize a proportionate distribution of these funds might disable this *program* in our less populous counties. The jurors said Failure to do this will continue to place a disproportionate *burden* on Fulton taxpayers.

Figure 1: Sample text from SemCor, br-a01 with the word *program* (word number 93) in consideration

The number in the brackets, by the side of the word, is its WordNet sense number. The numbers mentioned in the square brackets are the textual positions. For example, the word *law* appears in textual positions 66 and 72. The arrows going up-down show the *hyponymy* relations. Thus, 2 hyponyms of *sense number 1* of *cognition* are shown.

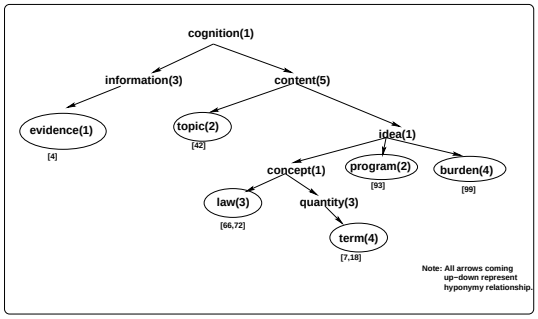


Figure 2: An extract of the WordNet graph, corresponding to the nouns underlined in figure 1

In the same way, one can consider the *holonymy*, *meronymy*, *synonymy* and *antonymy* relationships from the WordNet to capture all the nouns in a given piece of text. Consider the resultant WordNet subgraph. Also, suppose that distances are measured over edges, with every edge of unit distance and the distances are additive. Consider all the words that occur in the graph, within a distance of 4 from the 2^{nd} sense of the word *program*. We call the set of word-senses, within a fixed distance from the chosen synset as the *semantic set* corresponding to that synset. Fig. 3 is an example. The notations and the definitions are given in section 4.

program<80,2,<0,0,0,0,0,0>>, portion<95,1,<0,3,1,0,1,0>>, policy<58,2,<0,2,0,0,0,0>>, term<7,4,<0,3,1,0,0,0>>, topic<42,2,<0,1,2,0,0,0>>, law<66,1,<0,3,1,0,0,0>>, end<8,3,<0,4,0,1,0,0>>, term<18,4,<0,3,1,0,0,0>>, end<8,4,<0,1,0,1,0,1>>, practice<45,5,<0,1,3,0,0,0>>, manner<14,3,<0,4,1,0,0,0>>, law<66,3,<0,2,1,0,0,0>>, end<39,3,<0,4,0,1,0,0>>, end<39,4,<0,1,0,1,0,1>>, law<72,1,<0,3,1,0,0,0>>, burden<99,4,<0,1,1,0,0,0>>, city<59,1,<0,2,3,0,0,0>>, law<72,3,<0,2,1,0,0,0>>, city<31,1,<0,2,3,0,0,0>>, city<56,1,<0,2,3,0,0,0>>, evidence<4,1,<0,2,3,0,0,0>>

Figure 3: Semantic set corresponding to sense number 2 of the word *program*

4 Terminology

We want to find the correct senses of the words in a text T. Let W be a window in T having n nouns,

(w_1, w_2, w_3, ...w_n). For every w_i its s_i senses are σ_{i_1},.....$\sigma_{i_{s_i}}$. Let P_i be the position of w_i in the text.

Semantic Graph

Let G be that minimal sub-graph of the WordNet, which includes all the noun-senses σ_{i_k}, $1 \leq k \leq s_i$ and $1 \leq i \leq n$, from T. We call G, the *Semantic Graph* for the text T.

Let σ_{i_p} and σ_{j_q} be two noun-senses in the sub-graph G. Consider the shortest path from σ_{i_p} to σ_{j_q}. Let $\eta_1(\sigma_{i_p}, \sigma_{j_q})$, $\eta_2(\sigma_{i_p}, \sigma_{j_q})$, $\eta_3(\sigma_{i_p}, \sigma_{j_q})$, $\eta_4(\sigma_{i_p}, \sigma_{j_q})$, $\eta_5(\sigma_{i_p}, \sigma_{j_q})$ and $\eta_6(\sigma_{i_p}, \sigma_{j_{q_6}})$ respectively be the number of *hyponymy*, *hypernymy*, *meronymy*, *holonymy*, *synonymy* and *antonymy* arcs on this path.

Semantic vector

The *semantic vector* between two noun senses σ_{i_p} and σ_{j_q} in the graph G is the sequence

$< \eta_1(\sigma_{i_p}, \sigma_{j_q}),$ ..... $\eta_4(\sigma_{i_p}, \sigma_{j_q}),$ $\eta_5(\sigma_{i_p}, \sigma_{j_q}),$ $\eta_6(\sigma_{i_p}, \sigma_{j_q}) >$,

where the $\eta_i(\sigma_i, \sigma_{j_q})$'s are as given in previous definition. We denote the semantic vector by $N(\sigma_{i_p}, \sigma_{j_q})$.

Semantic distance

The concept of semantic distance has been explored in [15]. Broadly, two concepts of semantic distance have been mentioned there. They are *semantic similarity* and *semantic relatedness*. In this paper, we talk of semantic relatedness, as explored in [16]. But the measures of semantic distance that we adopt are little variants of what has been proposed by [16]. The first measure of the *semantic distance* of a noun-sense σ_{j_q} from σ_{i_p} in G corresponds to the minimum number of arcs that *must* be traversed in order to reach σ_{j_q} from σ_{i_p}.

From the fact that *hypernymy*, *hyponymy* and *meronymy*, *holonymy* are complementary, and that *synonymy* and *antonymy* are *symmetric*, it follows that the semantic distance is *commutative*.

The second measure of semantic distance will be given in section 5.2.

Semantic form

Recall the definition that P_i is the position of w_i in the text. The expression, $w_j < P_j, q, < N(\sigma_{i_p}, \sigma_{j_q} >>$ is called the *semantic form* for σ_{j_q} *with respect to* σ_{i_p}. We will denote it by $F(\sigma_{i_p}, \sigma_{j_q})$.

Semantic set

Consider every noun-sense σ_{j_q} in G, within a maximum semantic distance of R from σ_{i_p}. The collection of all the semantic forms $F(\sigma_{i_p}, \sigma_{j_q})$ is called the *semantic set* S_{i_p} for σ_{i_p}, with radius R. σ_{i_p} is called the *reference noun-sense* for S_{i_p}.

A semantic set S_{i_p} is of the form given in equation 1.

$$S_{i_p} = F(\sigma_{i_p}, \sigma_{i_p}), F(\sigma_{i_p}, \sigma_{i_{1_{p_1}}}) .... F(\sigma_{i_p}, \sigma_{i_{k_{p_k}}}) \quad (1)$$

where k is the length of the semantic set. For word w_i we have s_i semantic sets S_{i_1}, S_{i_2} , ... $S_{i_{s_i}}$. Also, for the word sense σ_{i_p} we define the position vector $\overline{P_{i_p}}$ and $\overline{M_{i_p}}$ as in equations 2 and 3.

$$\overline{P_{i_p}} =< P_{i_1} ... P_{i_k} > \quad (2)$$

$$\overline{M_{i_p}} =< N(\sigma_{i_p}, \sigma_{i_{1_{p_1}}}), ... N(\sigma_{i_p}, \sigma_{i_{k_{p_k}}}) > \quad (3)$$

An Example

Consider again the figure 1 which shows a sample from the text br-a01 of SemCor. The wordsenses of the underlined nouns in the text, form a semantic graph, part of which has been depicted in figure 2. For instance consider the word *program* which has position number 93 in br-a01 and *burden* which has position number 99 in br-a01. IN figure 2, it is shown that sense number 4 of *burden* and sense number 2 of *program* have the same hypernym - the sense number 1 of *idea*. Thus, the semantic distance between *program(2)* and *burden(4)* is 2. The semantic vector from *program(2)* to *burden(4)*, keeping *program(2)* as the reference word is $< \eta_1(\sigma_{93_2}, \sigma_{99_4})$,.. $\eta_6(\sigma_{93_2}, \sigma_{99_4}) >$=<1,1,0,0,0,0>

The distances traversed along the different relation arcs, in the figure 2 from *program(2)* to *burden(4)* are as given in the table 1.

The semantic form $F(\sigma_{93_2}, \sigma_{99_4})$ is given as *burden<99,4,<1,1,0,0,0,0>>*. σ_{99_4} is within a semantic distance of 4 from σ_{93_2}. The collection of all $F(\sigma_{93_2}, \sigma_{j_q}), 1 \leq q \leq s_j \forall$ words $w_j, j \neq i$ in the text T such that, σ_{j_q} is within a semantic distance 4 from σ_{93_2} is called the semantic set for σ_{93_2}, $S(\sigma_{93_2})$. This semantic set is given in figure 3.

5 The Approach

The problem of finding the appropriate sense for w_i can be transformed to the problem of choosing the corresponding appropriate semantic set for w_i. This means we intend to find a measure function $M(S_{i_p}) =$

Table 1: The distance along the different relation arcs, between *program(2)* and *buden(4)* as depicted in 2

Relation	Notation for dist.	Distance
hyponymy	$\eta_1(\sigma_{93_2}, \sigma_{99_4})$	1
hypenymy	$\eta_2(\sigma_{93_2}, \sigma_{99_4})$	1
meronymy	$\eta_3(\sigma_{93_2}, \sigma_{99_4})$	0
holonymy	$\eta_4(\sigma_{93_2}, \sigma_{99_4})$	0
synonymy	$\eta_5(\sigma_{93_2}, \sigma_{99_4})$	0
antonymy	$\eta_6(\sigma_{93_2}, \sigma_{99_4})$	0

m_{i_p} such that $\text{argmax}_{1 \le p \le s_i} M(S_{i_p})$ gives the correct sense for the word w_i.

The idea is that, a word-sense in the text indicates the presence of other word-senses in the piece of text in such a way that **semantically close word senses should also appear textually close**. Therefore, a word-sense in the text is affected by another word-sense in the text in two ways. First is that, *as the semantic distance between them increases, the influence should decrease*. Secondly, *as the textual distance between them increases, the influence should decrease.*

Intuitively, the first factor plays a predominant role in determining the sense of the word under consideration. This follows from the fact that **slight variation in textual position of a word-sense should not influence the sense of the passage as such. But a slight variation in semantic distance should considerably alter the sense of the passage.**

Based on these two observations, we state the hypothesis in section 5.1

5.1 Simple Manhattan measure

Hypothesis

The measure $M(S_{i_p})$ is of the form $M(\overline{P_{i_p}}, \overline{M_{i_p}})$. The contribution of each word σ_{i_j} in the semantic set to the score $M(S_{i_p})$ decreases exponentially its the semantic distance from w_i and decreases inversely with its textual distance from w_i.

Semantic distance (defined in section 4) can be restated as the *Manhattan distance*, $H(\sigma_{i_p}, \sigma_{j_q})$ in equation 4. Note that this measure, in contrast to the measure of semantic distance as given in [16], does not reduce the distance if the path connecting the two concepts changes 'direction too often'. (*e.g* of such a change is when the path connecting the two synsets, changes from say hypernymy to meronymy relation).

$$H(\sigma_{i_p}, \sigma_{j_q}) = \Sigma_{m=1}^{6} \eta_m(\sigma_i, \sigma_{j_q}) \tag{4}$$

According to the hypothesis mentioned above, the expression for the measure function is as given in equation 5.

$$M(\overline{P_{i_p}}, \overline{M_{i_p}}) = \Sigma_{F(\sigma_{i_p}, \sigma_{j_q}) \in S(\sigma_{i_p})} \frac{1}{|P_{j_q} - P_i|} \times e^{-H(\sigma_{i_p}, \sigma_{j_q})} \tag{5}$$

For a word w_i, the appropriate sense number is p and the second most appropriate sense number is $\overline{p}$ iff the conditions given in equations (6) and (7) are satisfied.

$$p = \underset{0 \le j \le s_i}{\text{argmax}} M(S_{i_j}) \tag{6}$$

$$\overline{p} = \underset{0 \le j \le s, j \ne k}{\text{argmax}} M(S_{i_j}) \tag{7}$$

5.2 Eucledian measure

Instead of using the *Manhattan distance*, one can use the *Eucledian distance* . The intuition is given in the figure 4

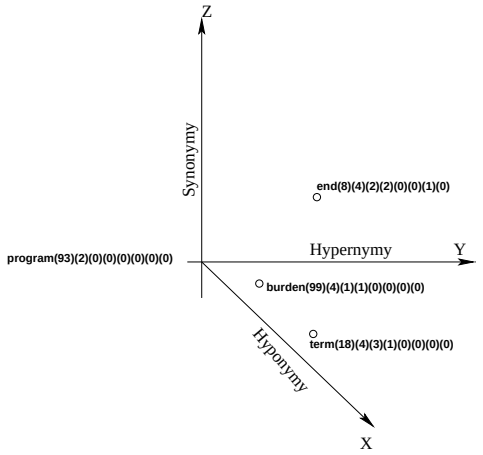


Figure 4: 3-D Graph showing the relative positions of three words with respect the the word *program*

We can look upon the words as being arranged in a six dimensional space, with each space corresponding to one of the 6 relations (*hypernymy etc*). The figure 4 for instance, shows the word-senses *end*(4), *burden*(4) and *term*(4), with respect to the word-sense *program*(2) in 3-D space of *hypernymy, hyponymy* and *synonymy.*

Instead of using the distance measure as in equation 4, we can use the measure $H(\sigma_{i_p}, \sigma_{j_q})$ as in equation 8. Again, this meaure of distance is different from that sugested in [16], because, instead of considering

change of direction along the path, we consider each of the 6 WordNet relations to be along orthogonal directions.

$$H(\sigma_{i_p}, \sigma_{j_q}) = \sqrt{\Sigma_{m=1}^{6}(\eta_m(\sigma_i, \sigma_{j_q})^2)} \quad (8)$$

The appropriate sense for the word w_i can be found as before, using equation 5 and 6. In the measure in equation 8, we give uniform weight-age to all the six relations - *hypernyms* etc. One can instead, give more weight-age to the *hypernymy* and *synonymy* relations as compared to the other relations (say, by taking cubes instead of squares), since, they determine the context of a passage of text, to a greater extent. This gives us the equations 9 and 10 for $H(\sigma_{i_p}, \sigma_{j_q})$.

$$E(\sigma_{i_p}, \sigma_{j_q}) = (\eta_2(\sigma_{i_p}, \sigma_{j_q})^3) + (\eta_5(\sigma_{i_p}, \sigma_{j_q})^3) \quad (9)$$

$$H(\sigma_{i_p}, \sigma_{j_q}) = \sqrt{\Sigma_{m=1,m\neq 2,5}^{6}(\eta_m(\sigma_i, \sigma_{j_q})^2) + E(\sigma_{i_p}, \sigma_{j_q})} \quad (10)$$

Again, one can employ equations 5 and 6 to find the appropriate sense for w_i.

Mutual Reinforcement

We may note that **a word w which has a unique sense in WordNet, helps disambiguate other words related to it**. That is, if word w_j has only one WordNet sense, we would like to give special attention to this information, in all the sets that contain σ_{j_1}. For instance, if the p^{th} semantic set for w_i, i.e S_{i_p} has the word w_j, with w_j having only onse sense in the WordNet, giving more weightage to w_j, sense number 1, will add additional emphasis on the p^{th} sense of w_i.

Moreover, we would like that this effect on σ_{i_p} be reflected on all the sets that contain σ_{i_p}.in turn. To ensure that this happens, we make the following changes to equation 5. Initially, we set the score for each semantic set to 1. Next, within the semantic sets for a word, we normalise the scores. Not that sets corresponding to unambiguous word senses (i.e word senses for the words having just one WordNet sense) will have a score of 1 initially. Then we find the new measure for each semantic set using equations 11 and 12:

$$I(\sigma_{i_p}, \sigma_{j_q}) = M(S_{j_q}) \times \frac{1}{|P_{j_q} - P_i|} \times e^{-H(\sigma_{i_p}, \sigma_{j_q})} \quad (11)$$

$$M(\overline{P_{i_p}}, \overline{M_{i_p}}) = \Sigma_{F(\sigma_{i_p}, \sigma_{j_q}) \in S(\sigma_{i_p})} I(\sigma_{i_p}, \sigma_{j_q}) \quad (12)$$

After updating all set measures for w_i using equation 12, we normalise the measures for the sets correspionding to w_i using equation 13.

$$M(S_{i_p}) = M(\overline{P_{i_p}}, \overline{M_{i_p}}) = \frac{M(S_{i_p})}{\Sigma_{r=1}^{s_i} M(S_{i_r})} \quad (13)$$

Note that in the equation 12 we have scaled the entry for each term σ_{j_q} in the set S_{i_p}, by the measure $M(S_{j_q})$ for the corresponding set S_{j_q}. This means that, if in a particular iteration, sense number q of w_j is found to be more probable than the other senses of w_j, then it's contribution to the scores of other sets is more than the other senses of w_j.

The pseudocode is summarised in figures 5 (INITIALISATION) and 6 (MUTUAL REINFORCEMENT).

1. **INTIALISATION**

2. Incrementally construct semantic chains S_{i_p}, $1 \leq p \leq i_s$, for each of the i_s WordNet senses of σ_i, $1 \leq i \leq n$.

3. for all $1 \leq i \leq n$ do

   (a) for all $1 \leq p \leq s_i$ do

      i. $M(S_{i_p}) = \frac{1}{s_i}$ /* Note that we have combined 2 steps into 1; setting $M(S_{ip})$ to 1 and then normalising */

Figure 5: The INITIALISATION Pseudocode for the method

6 Experiments and results

Experiments were performed over nouns in Brown corpus and checked against SemCor for correctness. As an example case, consider the 93^{rd} noun, *program* in the text in figure 1. It is tagged with sense number 2 in SemCor. Figure 7 shows the 8 semantic sets for the word *program*.

Using equations 4 and 5, we get the scores for the different sets as indicated by the bold number to the right of each set in the figure. *The scores stabilse after around 10 iterations.* We find highest score for the

1. do till the scores stabilise
   (a) **MUTUAL REINFORCEMENT**
   (b) for all $1 \leq i \leq n$ do
      i.
      ii. for all $1 \leq p \leq s_i$ do
         A. $H(\sigma_{i_p}, \sigma_{j_q}) = \Sigma_{m=1}^{6} \eta_m(\sigma_i, \sigma_{j_q})$ /*This could be replaced by the Eucledian measure.*/
         B. $M(\overline{P_{i_p}}, \overline{N_{i_p}}) = \Sigma_{F(\sigma_{i_p}, \sigma_{j_q}) \in S(\sigma_{i_p})} M(S_{j_q}) \times \frac{1}{|P_{j_q} - P_i|} \times e^{-H(\sigma_{i_p}, \sigma_{j_q})}$
   (c) **NORMALISATION**
   (d) for all $1 \leq i \leq n$ do
      i. for all $1 \leq p \leq s_i$ do
         A. $M(S_{i_p}) = M(\overline{P_{i_p}}, \overline{N_{i_p}}) = \frac{M(S_{i_p})}{\Sigma_{r=1}^{s_i} M(S_{ir})}$

Figure 6: The MUTUAL REINFORCEMENT Pseudocode for the method

program<93,1,<0,0,0,0,0,0>>, ... evidence<4,1,<0,2,2,1,0,0>> = 0.312, **0.023**
program<93,2,<0,0,0,0,0,0>>, ... portion<95,1,<0,3,1,0,1,0>> = 1.129, **0.088**
program<93,3,<0,0,0,0,0,0>>, ... election<35,2,<0,3,1,0,1,0>> = 0.144, **0.009**
program<93,4,<0,0,0,0,0,0>>, ... report<24,1,<0,1,1,0,0,0>> = 0.899, **0.611**
program<93,5,<0,0,0,0,0,0>>, ... title<65,1,<0,1,1,1,0,0>> = 0.186, **0.017**
program<93,6,<0,0,0,0,0,0>>, . distribution<91,3,<0,1,3,1,0,0>> = 1.107, **0.053**
program<93,7,<0,0,0,0,0,0>>, ... laws<38,1,<0,2,2,0,0,1>> = 0.629, **0.045**
program<93,8,<0,0,0,0,0,0>>, ... city<56,1,<0,1,3,0,0,1>> = 0.473, **0.022**

Figure 7: Example of 8 semantic sets for the word *program*

second set - thus indicating *sense number 2*. Thus, as per our expectation, the algorithm correctly disambiguated the word *program*. On the other hand, using equations 8 and 5, we get the scores as the underlined number, to the right of each set, in the figure 7. As far as the *Eucledian distance* was concerned, it did not make a big difference, whether we used the measure as suggested in equation 8 or 10. The experiments were carried out on the fist 100 nouns for each of 10 documents from the *Brown corpus*. 2 tests were done - (1) comparing the top ranked sense p and (2) comparing the 2 top ranked senses, p and $\overline{p}$ derived using equation 6. The results for 5 of them are tabulated below.

The average *precision* obtained using the *Eucledian measure* was $3-4\%$ lower than that obtained using

Table 2: Results with top sense for each of 10 *brown corpus* documents

Text	Coverage (%)	Precision (%)	Recall (%)
a01	99	70	69.3
a02	98	69	67.6
a11	96	63	60.5
a12	95	65.0	61.8

Table 3: Results with top 2 senses for each of 10 *brown corpus* documents

Text	Coverage (%)	Precision (%)	Recall (%)
a01	99	83.8	83.0
a02	98	75.5	74.0
a11	96	79.2	76.0
a12	95	74.7	71.0

the *Manhattan measure*. The comparison of our algorithm was done with [4], one of the best known *Unsupervised* WSD algorithms.The comparison was performed on the entire text of *br-a01*. The results were as mentioned in table 4

Table 4: Comparison with [4]

	Aigrre		Our algo	
	precision	recall	precision	recall
br-a01	66.4	58.8	76.9	68.2
br-a02	-	-	70.9	68.8
br-b13	-	-	77.8	75.5
br-c04	-	-	67.3	64.10

The window size $|W| = n$, for all the above tests was chosen as 100. Changing it to 150 produced improvement by $5-7\%$.

7 Conclusions

The algorithm discussed in this paper is unsupervised. Currently, it is designed only for disambiguating nouns. All it needs is WordNet, an extensively used lexical database. It can disambiguate any free running text, provided that the *part of speech tags* are provided. The idea behind the algorithm is theoretically well supported. It has many special features compared to previous unsupervised algorithms. Even though a window of words is used for disambiguation, all the nouns in the window are not considered with equal importance for disambiguating a word in the

text - the importance decreases with increasing distance in the text as well as with increasing *Manhattan* or *Eucledian* distance in the WordNet. Also note that the same word, occurring in different parts of the window is disambiguated in a different way - it considers separately, the multiple occurrences of same word in the same window.

With slight modification, this algorithm can be used for disambiguating *verbs*, *adjectives* in any text. The corresponding *verb* and *adjective* taxonomies in the WordNet can be used for these purposes - in a most similar way.

The algorithm can be improved by choosing a different measure function or choosing different measures of semantic distance, than the two mentioned in this paper. Also, consideration of collocation of words and verb-noun collocations, should give additional clues for disambiguation.

References

[1] David Yarowsky. "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora", In *Proceedings of the* 14^{th} *International Conference on Computational Linguistics (COLING-92)*, pages 454-460, Nantes, France, 1992.

[2] Hwee Tou Ng and Hian Beng Lee, "Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach", In *Proceedings of ACL96, 1996.*

[3] Adam Kilgarriff, "Gold Standard Data-sets for Evaluating Word Sense Disambiguation Programs", *Computer Speech and Language 12 (4), Special Issue on Evaluation*, 1998.

[4] Agirre.E and Rigau.G, "Word sense disambiguation using conceptual density", *Proceedings of COLING'96.*

[5] Phil Resnik, "Selectional preference and sense disambiguation", *Proceedings of ANLP-97 Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington D.C., USA, 1997.

[6] Hinrich Schtze, "Automatic Word Sense Discrimination", *Computational Linguistics*, Volume 24, Number 1, 1998.

[7] Nancy Ide, "Parallel Translations as Sense Discriminators", *Proceedings of SIGLEX99*, Washington D.C, USA, 1999.

[8] Green, S.J., "Automatically Generating Hypertext by Computing Semantic Similarity" *Ph.D. Thesis, University of Toronto*, 1997.

[9] Fellbaum, Christiane, ed., "WordNet: An Electronic Lexical Database" *MIT Press*, May 1998.

[10] Cowie.J, Guthrie.L and Guthrie.J, *L*exical Disambiguation using simulated annealing. Proceedings of the 5th International Conference on Computational Linguistics. COLING-93 (1992), pp157-161.

[11] McRoy.S, *U*sing multiple knowledge sources for Word Sense Disambiguation. Computational Linguistics 18.1 (1992), pages 1-30.

[12] Li.X, Szpakowicz.S and Matwin.M, *A* WordNet based algorithm for word sense disambiguation. Proceedings of the 14^{th} Joint International Conference on Artificial Intelligence.

[13] Resnik.P, *S*electional preference and sense disambiguation. In proceedings of the ACL Singlex Workshop on Tagging Text with Lexical Semantics, Why, What and How? (Washington DC, April 1997).

[14] Miller.G, Chodorow.M, Landes.S, Leacock.C and Thomas.R, *U*sing a semantic concordance for sense indentification. Proceedings of the 37^{th} Annual Meeting of the Association for Computational Linguistics (ACL-99).

[15] Alexander Budanitsky, Graeme Hirst, *S*emantic Distance in WordNet: An experimental, application-oriented evaluation of five measures Proceedings of WordNet and Other Lexical Resources Workshop, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, June 2001.

[16] Graeme Hirst, David St-Onge, *L*exical chains as representations of context for the detection and correction of malapropisms, In: Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA: The MIT Press, 1998, 305–332.

Towards Sense-Disambiguated Association Thesauri

Hiroyuki Kaji and Yasutsugu Morimoto

Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
{kaji, y-morimo}@crl.hitachi.co.jp

Abstract

We developed a method for generating a sense-disambiguated association thesaurus, in which word senses are distinguished according to the related words, from a bilingual comparable corpus. The method aligns pairs of related words translingually by looking up a bilingual dictionary. To overcome both the problem of ambiguity in the translingual alignment of pairs of related words and that of disparity of topical coverage between corpora of different languages, we devised an algorithm for calculating the correlation between the senses of a polysemous word and its related words iteratively according to the set of words related to both the polysemous word and each of the related words. A preliminary experiment using Wall Street Journal and Nihon Keizai Shimbun corpora demonstrated that the method produces a sense-disambiguated association thesaurus successfully. We expect the sense-disambiguated association thesaurus will play essential roles in information retrieval and filtering. Namely, it enables word sense disambiguation of documents and queries as well as effective query expansion. It also functions as an effective user interface for translingual information retrieval.

1 Introduction

An association thesaurus, that is, a collection of pairs of related words, plays an essential role in information retrieval. Query expansion using a corpus-dependent association thesaurus improves recall and/or precision (Jing and Croft 1994; Schuetze and Pedersen 1994; Mandala et al. 1999). Navigation in an association thesaurus allows users to efficiently explore information through a large text corpus even when their information needs are vague (Kaji et al. 2000).

Association thesauri have the advantage of being possibly generated from corpora automatically. However, they have a drawback that they cannot distinguish between the senses of a polysemous word; namely, although each word that is related to a polysemous word is usually relevant to a specific sense of the polysemous word, the association thesauri list all related words regardless of sense. Query expansion using words irrelevant to the sense of user's interest decreases the precision of retrieval. A mixed list of related words relevant to different senses of a polysemous word prevents users from navigating smoothly in the association thesaurus.

In order to solve this problem, we propose a method for generating a sense-disambiguated association thesaurus, in which the senses of a polysemous word are distinguished. More specifically, the words related to a polysemous word are classified according to the sense of the polysemous word to which they are relevant.

2 Approach

The high cost of sense-tagging a corpus prohibits us from collecting pairs of related "senses" directly from a corpus. Accordingly, we adopt a strategy to extract pairs of related "words" from a corpus and then transform each of them to a pair of related senses. This transformation is done through translingual alignment of pairs of related words, as shown in Figure 1. The underlying assumptions are:

(1) The senses of a polysemous word in a language are lexicalized differently in another language (Resnik and Yarowsky 2000).
(2) Translations of words that are related in one language are also related in the other language (Rapp 1995).

According to the first assumption, we define each sense of a polysemous word x of the first language by a synonym set consisting of x itself and one or more of its translations $y_1, y_2, \ldots$ into the second language. The synonym set is similar to that in WordNet (Miller 1990) except that it is bilingual, not monolingual. Examples of some sets are given below.

{tank, タンク<*TANKU*>, 水槽<*SUISO*>, 槽<*SO*>}

{tank, 戦車<*SENSHA*>}

These synonym sets define the "container" sense and the "military vehicle" sense of "tank" respectively.

According to the second assumption, our method aligns first-language pairs of related words with second-language pairs of related words via a bilingual dictionary. An alignment of a first-language pair of a polysemous word and its related word with its counterpart in the second language is transformed into a pair of a sense of the polysemous word and a clue. A word related to the polysemous word is called a clue, because it helps to determine the sense of the polysemous word. For example, the alignment of (tank, gasoline) with (タンク<*TANKU*>, ガソリン<*GASORIN*>) results in a sense-clue pair ({tank, タンク<*TANKU*>, 水槽<*SUISO*>, 槽<*SO*>}, gasoline), and the alignment of (tank, soldier) with (戦車<*SENSHA*>, 兵士<*HEISHI*>) results in a sense-clue pair ({tank, 戦車<*SENSHA*>}, soldier).

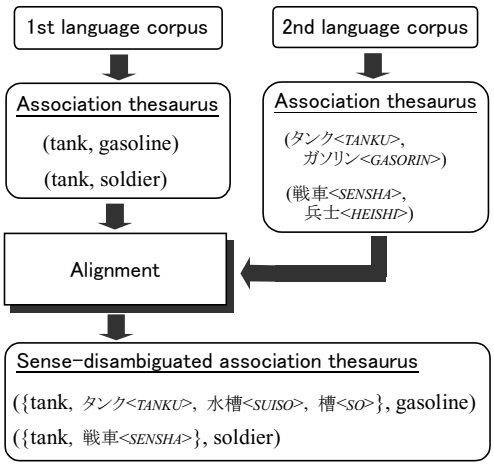


Figure 1: Proposed framework for producing a sense-disambiguated association thesaurus

3 Proposed Method

3.1 Problems and solution

In the framework of aligning pairs of related words translingually, we encounter two major problems: the ambiguity in alignment of pairs of related words, and the disparity of topical coverage between the corpora of the two languages. The following subsections discuss how to overcome these problems.

3.1.1 Coping with ambiguity in alignment

Matching of pairs of related words via a bilingual dictionary often suggests that a pair in one language can be aligned with two or more pairs in the other language (Dagan and Itai 1994; Kikui 1998). To cope with this ambiguity, we evaluate the plausibility of alignments according to the following two assumptions.

(a) Correct alignments are those with pairs of strongly related words.
(b) Correct alignments are accompanied by a lot of common related words that can be aligned with each other.

Then, according to the plausibility of alignments, we calculate the correlation between the senses of a polysemous word and the clues, i.e., words related to the polysemous word.

To precisely estimate the plausibility of alignments according to assumption (b), we should use the correlation between senses and clues. Therefore, we developed an algorithm for calculating the correlation between senses and clues iteratively (see Subsection 3.2.2 for details).

3.1.2 Coping with disparity between corpora

Matching of pairs of related words via a bilingual dictionary often results in a number of pairs not being aligned with any pair. One reason for this is the disparity of topical coverage between the corpora of two languages; another reason is the insufficient coverage of the bilingual dictionary.

To make it possible to acquire the correlations between senses and a clue, even from a first-language pair of related words that cannot be aligned with any second-language pair of related words, we introduce a "wild card" pair. The wild-card pair is a virtual pair related to every word of the second language and implies every sense of the polysemous word of the first language. When a pair cannot be aligned with any other pair, we align it with the wild-card pair compulsorily. We apply the iterative algorithm mentioned in Subsection 3.1.1 to all alignments including alignments with the wild-card pair. Although an alignment with the wild-card pair produces no distinction among the senses of the polysemous word in the first iteration, it produces distinction after the second iteration (An example is given in Section 3.3).

3.2 Algorithm

Our method consists of two steps: translingual alignment of pairs of related words and iterative calculation of correlation between senses and clues. The following subsections give a detailed description of these steps.

3.2.1 Alignment of pairs of related words

An association thesaurus is a collection of pairs of related words with a measure of association between them. In this section, R_X and R_Y denote association thesauri of the first and second languages, respectively. We use mutual information, which is calculated according to co-occurrence statistics, as a measure of association; $MI(x,x')$ denotes the mutual information value of a pair of related words (x,x') ($\in R_X$), and $MI(y,y')$ denotes that of a pair of related words (y,y') ($\in R_Y$), respectively. It should be noted that the measure of association is not limited to the mutual information.

Alignments of pairs of related words between R_X and R_Y, each of which is accompanied by a set of common related words, are extracted through the following procedure.

(1) Extraction of possible alignments

First, for each polysemous word x of the first language, we extract the clue set $X(x)$, which is defined as the set of words related to x, i.e.,

$$X(x)=\{x' \mid (x,x') \in R_X\}.$$

Henceforth, we denote the j-th clue of x as $x'(j)$. Then, for each pair of x and $x'(j)$ ($\in X(x)$), we extract the counterpart set $Y(x, x'(j))$, which is defined as the set of second-language pairs with which the first-language pair $(x, x'(j))$ is possibly aligned, i.e.,

$$Y(x, x'(j))=\{ (y, y') \mid (y, y') \in R_Y, (x, y) \in D, (x'(j), y') \in D\}.$$

Where D denotes a bilingual dictionary, i.e., a collection of pairs consisting of a first-language word and a second-language word that are translations of each other.

(2) Extraction of sets of common related words

(a) In case the counterpart set $Y(x, x'(j))$ is nonempty, for each alignment of $(x, x'(j))$ with (y, y') ($\in Y(x, x'(j))$), we extract a set of common related words $Z((x, x'(j)), (y, y'))$, which is defined as a set of first-language words related to the first-language pair $(x, x'(j))$ and with at least one translation related to the second-language pair (y, y'), i. e.,

$$Z((x, x'(j)), (y, y'))=\{x'' \mid (x, x'') \in R_X, (x'(j), x'') \in R_X\} \cap \{x'' \mid \exists y'' (x'', y'') \in D, (y, y'') \in R_Y, (y', y'') \in R_Y\}.$$

(b) In case the counterpart set $Y(x, x'(j))$ is empty, or the set of common related words $Z((x, x'(j)), (y, y'))$ extracted in the step (a) is empty for all counterparts (y, y') ($\in Y(x, x'(j))$), we align the first-language pair $(x, x'(j))$ with the wild-card pair (y_0, y_0') and construct a set of common related words as follows:

$$Z((x, x'(j)), (y_0, y_0'))=\{x'' \mid (x, x'') \in R_X, (x'(j), x'') \in R_X\}.$$

3.2.2 Calculation of correlation between senses and clues

We define the correlation between each sense of a polysemous word and a clue as the mutual information between them multiplied by the maximum plausibility of alignments that imply the sense. That is,

$$C_n(S(i),x'(j)) = MI(x,x'(j)) \cdot \frac{\max_{y(\in S(i)\cup\{y_0\}),y'} \left(MI(y,y') \cdot \sum_{x'' \in Z((x,x'(j)),(y,y'))} C_{n-1}(S(i),x'') \right)}{\max_k \left\{ \max_{y(\in S(k)\cup\{y_0\}),y'} \left(MI(y,y') \cdot \sum_{x'' \in Z((x,x'(j)),(y,y'))} C_{n-1}(S(k),x'') \right) \right\}},$$

where n denotes the iteration number, and $S(i)$ denotes the i-th sense of the polysemous word x, precisely, the synonym set that defines the i-th sense of x.

The numerator of the second term in the above formula is the maximum of plausibility of alignments that imply the sense, and the denominator is introduced to normalize the plausibility of alignments. The first term of the plausibility of alignment, the mutual information of the second-language pair of related words, corresponds to assumption (a) in Subsection 3.1.1. We assign an arbitrary value larger than zero to the mutual

Alignment	Set of common related words	Sense(s) implied
((tank, troop), (水槽<*SUISO*>, 群れ<*MURE*>))	{air, area, fire, government}	{tank, タンク<*TANKU*>, 水槽 <*SUISO*>, 槽<*SO*>}
((tank, troop), (槽<*SO*>, 多数<*TASU*>))	{area, army, control, force}	
((tank, troop), (戦車<*SENSHA*>, 群<*GUN*>))	{area, army, battle, commander, force, government}	{tank, 戦車<*SENSHA*>}
((tank, troop), (戦車<*SENSHA*>, 多数<*TASU*>))	{Serb, area, army, battle, force, government}	
((tank, troop), (戦車<*SENSHA*>, 隊<*TAI*>))	{Russia, Serb, air, area, army, battle, commander, defense, fight, fire, force, government, helicopter, soldier}	
((tank, gallon), wild card)	{Ford, Institute, car, explosion, fuel, gas, gasoline, leak, natural-gas, oil, pump, toilet, treaty, truck, vehicle, water}	{tank, タンク<*TANKU*>, 水槽 <*SUISO*>, 槽<*SO*>}, {tank, 戦車<*SENSHA*>}

(a) Alignments and accompanying sets of common related words

information of the wild-card pair (y_0, y_0'). Note that the value of the mutual information of the wild-card pair does not have an effect on the results. The second term of the plausibility of alignment, the sum of the correlations between the sense and the common related words, corresponds to assumption (b) in Subsection 3.1.1.

We set the initial values of the correlations between senses and clues as follows:

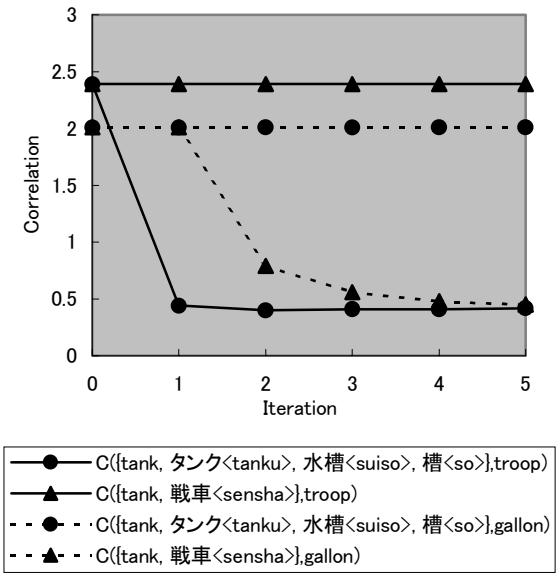
$$C_0(S(i), x'(j)) = MI(x, x'(j)).$$

In the present implementation, we iterate the calculation five times, which makes the correlation values converge. The iteration results in a correlation matrix between the senses of the polysemous word x and the clues. We do not determine the only sense that each clue suggests, but leave using the sense-vs.-clue correlation matrix to application systems.

3.3 Example of calculation

An example of calculating sense-vs.-clue correlations for an English polysemous word "tank" is shown in Figure 2. An English pair of related words (tank, troop) is aligned with five Japanese pairs of related words (水槽<*SUISO*>, 群れ<*MURE*>), (槽<*SO*>, 多数<*TASU*>), (戦車<*SENSHA*>, 群<*GUN*>), (戦車 <*SENSHA*>, 多数<*TASU*>), and (戦車<*SENSHA*>, 隊<*TAI*>). The five sets of common related words that accompany these alignments are shown in Figure 2(a). On the contrary, another English pair of related words (tank, gallon) cannot be aligned with any Japanese pair of related words and, therefore, is aligned with the wild-card pair. The set of common related words that accompanies the alignment of (tank, gallon) with the wild-card pair is also shown in Figure 2(a).

Figure 2(b) shows how the correlation values between the senses of "tank" and the two clues "troop" and "gallon" converge. The correlations with irrelevant senses approach certain small values as the iteration proceeds, while the correlations with relevant senses are kept constant. Note that the correlation value between {tank, タンク<*TANKU*>, 水槽 <*SUISO*>, 槽<*SO*>} and "gallon" and that between {tank, 戦車 <*SENSHA*>} and "gallon", both of which are based on the alignment with the wild-card pair, begin to diverge after the second iteration.



(b) Convergence of correlations

Figure 2: Example of calculating sense-vs.-clue correlations

4 Experiment

4.1 Experimental method

We conducted an experiment to study the feasibility of our method. In this experiment, the first and second languages were English and Japanese, respectively.

First, input data were prepared as follows.

(i) Association thesauri

An English association thesaurus was generated from a Wall Street Journal corpus (July, 1994 to Dec., 1995; 189 Mbytes), and a Japanese association thesaurus was generated from a Nihon Keizai Shimbun corpus (Dec., 1993 to Nov., 1994; 275 Mbytes). The procedure used is outlined as follows (Kaji et al. 2000). Mutual information was calculated for each pair of words according to the frequency of co-occurrence in a window, and pairs of words having a mutual information value larger than a threshold were selected. The words were restricted to nouns and unknown words, which are probably nouns. The size of the window was set to 25 words excluding function words, and the threshold of mutual information value was set to 0.

(ii) Test words

60 English polysemous nouns, whose different senses appear in newspapers, were selected as the test words, and their senses were defined by using their translations into Japanese. The frequencies of the test words in the corpus ranged from 39,140 ("share", the third noun in descending order of frequency) to 106 ("appreciation", the 2,914th noun).

The number of senses defined per test word ranged from 2 to 8, and the average was 3.4.

(iii) Bilingual dictionary

An English-Japanese noun dictionary was compiled from the EDR (Japan Electronic Dictionary Research Institute) English-to-Japanese and Japanese-to-English dictionaries. The resulting dictionary included 269,000 English nouns and 276,000 Japanese nouns.

Then, a sense-vs.-clue correlation matrix was produced for each test word by the method described in Section 3. Finally, the clues were classified according to their correlation with the senses. Namely, the sense having the largest correlation value was selected for each clue on the assumption that a clue is relevant to only one sense (Yarowsky 1993). Although this assumption is not always true, we did so because it is most important to distinguish the most relevant sense from the others.

4.2 Experimental results

Table 1(a) is a classified list of clues obtained for a test word "tank", and Table 1(b) is that obtained for another test word "intelligence". In these lists, clues are sorted in descending order of a score, which is defined as the minimum difference between the correlation with the sense and those with the other senses, i.e.,

$$Score(c) = \min_{S' \neq S} \left[C_5(S,c) - C_5(S',c) \right],$$

where *Score(c)* denotes the score of a clue *c* in the list corresponding to a sense *S*. The score indicates the capability of the clue distinguishing the most relevant sense from the others.

Note that Table 1 lists the top 50 clues for each sense. The total number of clues obtained for each sense of "tank" was as follows:

{tank, タンク<*TANKU*>, 水槽<*SUISO*>, 槽<*SO*>}: 86

{tank, 戦車<*SENSHA*>}: 89

As for "intelligence", two senses were defined: the "ability to learn" sense and the "information" sense. The total number of clues obtained for each sense was as follows:

{intelligence, 知能<*CHINO*>, 知性<*CHISEI*>}: 64

{intelligence, 情報<*JOHO*>, 諜報<*CHOHO*>}: 153

The experiment demonstrated the effectiveness of our method. At the same time, it revealed a few problems. First, when it happens that the second-language association thesaurus includes one or more counterparts of a first-language pair of related words but all of them are incorrect ones, the method causes an error. A sense-clue pair ({tank, タンク<*TANK*>, 水槽<*SUISO*>, 槽<*SO*>}, Poland) included in Table 1(a) is an example. The Japanese association thesaurus included an incorrect counterpart of (tank, Poland), i.e., (水槽<*SUISO*>, 波<*NAMI*>), but it did not include any correct counterpart of (tank, Poland), e.g., (戦車<*SENSHA*>, ポーランド<*PORANDO*>). Consequently, (tank, Poland) was aligned only with (水槽<*SUISO*>, 波<*NAMI*>), which resulted in the incorrect sense-clue pair.

Second, the experimental results show that it is difficult to distinguish a generic or non-topical sense from the other senses. An example is given below. Three senses of "measure" were defined: the "amount, size, weight, etc." sense, the "action taken to gain a certain end" sense, and the "law" sense. The number of clues obtained for each sense was as follows:

{measure, 量<*RYO*>, 尺度<*SHAKUDO*>, 指数<*SHISU*>}: 39

{measure, 対策<*TAISAKU*>, 手段<*SHUDAN*>, 処置<*SHOCHI*>}: 1

{measure, 法案<*HOAN*>, 議案<*GIAN*>, 法令<*HOREI*>}: 93

The method failed to obtain effective clues for selecting the second sense, which is extremely generic, although "measure" in this sense occurred frequently in the corpus.

5 Future Extensions

5.1 From sense-vs.-clue correlation to sense-vs.-sense correlation

The sense-vs.-clue correlation matrix is an intermediate form of sense-disambiguated association thesaurus. It should be transformed further into a sense-vs.-sense correlation matrix. This transformation can be done straightforwardly.

Let's take a pair of related words (tank, troop) as an example. The sense-vs.-clue correlation matrix produced for a polysemous word "tank", which is denoted as *M*(tank), includes the following pairs of a sense and a clue.

({tank, タンク<*TANKU*>, 水槽<*SUISO*>, 槽<*SO*>}, troop)

({tank, 戦車<*SENSHA*>}, troop)

Likewise, the sense-vs.-clue correlation matrix produced for another polysemous word "troop", which is denoted as *M*(troop), includes the following pairs of a sense and a clue.

({troop, 群れ<*MURE*>, 群<*GUN*>, 多数<*TASU*>}, tank)

({troop, 軍隊<*GUNTAI*>, 隊<*TAI*>, 部隊<*BUTAI*>}, tank)

So a pair of senses is produced by combining two pairs of a sense and a clue, one from *M*(tank) and the other from *M*(troop). The correlation value of the pair of senses is defined as the minimum of the correlation values of the combined pairs of a sense and a clue. For example,

C({tank, 戦車<*SENSHA*>}, {troop, 軍隊<*GUNTAI*>, 隊<*TAI*>, 部隊<*BUTAI*>}) = min [*C*({tank, 戦車<*SENSHA*>}, troop), *C*({troop, 軍隊<*GUNTAI*>, 隊<*TAI*>, 部隊<*BUTAI*>}, tank)].

5.2 Use of syntactic co-occurrence

We have conducted another experiment to evaluate word sense disambiguation using the sense-vs.-clue correlation matrix, which will be reported in detail at another opportunity. Although the overall results have been promising, our method has its limitations.

The present method deals with only nouns, and it extracts clues for word sense disambiguation according to co-occurrence in a window. However, it is obvious that doing this is not suitable for all polysemous words. Syntactic co-occurrence is more useful for disambiguating some sorts of polysemous words (Lin 1997). It is an important and interesting research issue to extend our method so that it can extract clues according to syntactic co-occurrence. This extended method does not replace the present method; however, we should combine both methods or use the one suitable for each polysemous word.

The framework of our method is compatible with syntactic co-occurrence. Basically, we only have to incorporate a parser into the association thesaurus generator. A parser of the first language is indispensable, but a parser of the second language is not. As for the second language, we may use co-occurrence in a small-sized window instead of syntactic co-occurrence.

6 Discussion

(a) List of clues relevant to each sense of "tank"

{tank, タンク<*TANKU*>, 水槽<*SUISO*>, 槽<*SO*>} *		{tank, 戦車<*SENSHA*>} **	
Clue	Score	Clue	Score
Walbro	5.13	artillery	4.04
ammonia	4.83	Grozny	2.98
static electricity	4.45	commander	2.65
Mrs. Tramm	4.15	Chechen	2.63
gasket	4.13	Chechnya	2.56
Jon-Luke	3.91	Mr. Yeltsin's	2.54
vapor	3.85	Patton	2.43
fuel tank	3.74	Serb	2.42
Aruba	3.55	Bosnian government	2.40
Zeus	3.24	missile	2.28
kangaroo	3.24	Cutiron	2.27
fuel	2.95	ball	2.17
pickup truck	2.87	treaty	2.17
leak	2.76	Yeltsin's	2.16
toilet	2.74	ammunition	2.14
tank barge	2.61	Polish method	2.03
fish	2.56	helicopter	2.01
Spar	2.43	soldier	2.00
tide	2.42	Mr. Gaffney	1.97
truck	2.34	Gaffney	1.95
pump	2.26	troop	1.92
liquid	2.25	thud	1.87
underground	2.24	weapon	1.84
Pena	2.23	civilian	1.82
concrete	2.22	Belarus	1.80
pickup	2.21	assault	1.73
gasoline	2.19	Bosnian	1.71
static	2.17	method	1.71
float	2.12	rebel	1.70
ozone	2.05	Yeltsin	1.68
temperature	1.94	NATO	1.66
recall	1.93	Mr. Yeltsin	1.64
electricity	1.90	parliament	1.51
tank car	1.85	Russian	1.48
plastic	1.84	army	1.39
explosion	1.82	U.N.	1.33
GM	1.78	bomb	1.25
rush	1.76	Army	1.25
safety	1.73	Polish	1.19
Poland	1.71	military	1.17
Mercedes	1.69	Rutkowski	1.15
emission	1.68	Pentagon	1.11
barge	1.60	defense	1.09
gallon	1.55	battle	1.07
design	1.46	force	1.05
fragment	1.42	Progress	1.02
bottom	1.39	Heritage Foundation	1.00
road	1.39	ton	1.00
Shell	1.35	column	0.97
blue	1.30	Force	0.92

* a large container for storing liquid or gas
** an enclosed heavily armed, armored vehicle

(b) List of clues relevant to each sense of "intelligence"

{intelligence, 知能<*CHINO*>, 知性<*CHISEI*>} ***		{intelligence, 情報<*JOHO*>, 諜報<*CHOHO*>} ****	
Clue	Score	Clue	Score
trait	3.76	CIA	5.19
curve	3.43	spy	4.55
domain	3.03	mole	4.49
secret	1.89	Pyongyang	3.12
shoot	1.88	U.S. military	3.10
consequence	1.78	palace	3.01
Hamlet	1.73	Directorate of Operation	2.91
Mainstream Science	1.67	intelligence budget	2.75
human	1.60	secret service	2.75
community	1.50	rod	2.74
domain name	1.50	satellite	2.61
capability	1.49	double agent	2.52
understanding	1.47	Defense Intelligence Agency	2.45
outcome	1.44	Woolsey	2.44
writer	1.43	Deutch	2.39
conclusion	1.42	U.S. intelligence	2.38
score	1.39	agent	2.37
IQ test	1.28	Intelligence Committee	2.35
book	1.28	Shalikashvili	2.32
IQ	1.27	intelligence community	2.31
author	1.26	Mr. Deutch	2.31
analysis	1.20	intelligence agency	2.27
knowledge	1.12	Kalugin	2.26
difference	1.07	weapon	2.25
Bell Curve	1.02	Mr. Woolsey	2.23
story	0.96	defector	2.19
study	0.95	intelligence service	2.17
child	0.93	Ames	2.11
test	0.90	espionage	2.09
Curve	0.89	Aspin	2.01
psychologist	0.88	Torricelli	1.98
society	0.88	analyst say	1.98
Mainstream	0.81	Seoul	1.98
woman	0.79	policy maker	1.97
research	0.71	Serb	1.91
white	0.67	assertion	1.90
academic	0.65	TI	1.89
fluid	0.64	fraction	1.81
tool	0.63	terrorism	1.81
life	0.63	annual budget	1.79
extreme	0.62	North Korean	1.78
Murray	0.59	KGB	1.73
gathering	0.59	State Department	1.70
man	0.57	military service	1.70
way	0.51	middle	1.69
view	0.49	Mr. Wolf	1.65
Science	0.47	East German	1.64
good	0.45	launder	1.64
discussion	0.42	Defense	1.64
source	0.39	Cold War	1.63

*** ability to learn, reason, and understand
**** information about an enemy

Table 1: Excerpt from the produced sense-disambiguated association thesaurus

6.1 Usefulness of sense-disambiguated thesaurus

The usefulness of the sense-disambiguated association thesaurus for information retrieval and filtering is discussed below. First, when it is shared by a system and users, the sense-disambiguated association thesaurus enables users to input

unambiguous queries. The system does not need to sense-disambiguate queries, since they are already disambiguated.

Second, the sense-disambiguated association thesaurus definitely improves the performance of query expansion. Because it enables a query to be expanded with related words relevant to the sense of user's interest, not with related words regardless of sense.

Third, the sense-disambiguated association thesaurus can be effectively used to sense-disambiguate documents. The sense of a polysemous word in a document is determined by comparing the context with the clues of each sense.

Finally, the sense-disambiguated association thesaurus, in which a sense is defined by a set of bilingual synonyms, functions as a user interface for translingual information retrieval. A user, who may not understand the second language, recognizes senses via the clues of the first language, and the system obtains second-language translation(s) from the synonym set specified by the user.

6.2 Word sense disambiguation and bilingual corpora

Word sense disambiguation using bilingual corpora has an advantage in that it enables unsupervised learning. However, the previous methods, which align instances of words (Brown et al. 1991), require a parallel corpus and, therefore, are applicable to limited domains. On the other hand, our new method requires a comparable corpus. The comparability required by the new method is very weak: any combination of corpora of different languages in the same domain, e.g., Wall Street Journal and Nihon Keizai Shimbun, is acceptable as a comparable corpus. Thus the new method has an advantage over the previous methods in being applicable to many domains.

Word sense disambiguation using bilingual corpora has a limitation because the senses of a first-language polysemous word are not always lexicalized differently in the second language. Second-language translations that preserve the ambiguity cause erroneous disambiguation. To avoid this problem, we eliminate translations that preserve the ambiguity from the synonym sets defining senses.

An example is given below.

{title, 肩書き<*KATAGAKI*>, 称号<*SHOGO*>, ~~タイトル<*TAITORU*>~~, 敬称<*KEISHO*>}

{title, 題名<*DAIMEI*>, 題目<*DAIMOKU*>, 表題<*HYODAI*>, 書名<*SHOMEI*>, ~~タイトル<*TAITORU*>~~}

{title, ~~タイトル<*TAITORU*>~~, 選手権<*SENSHUKEN*>}

These synonym sets define three senses of "title", the "person's rank or profession" sense, the "name of a book or play" sense, and the "championship" sense. A Japanese translation "タイトル<*TAITORU*>", which represents all these senses, is eliminated from all these synonym sets.

The method of eliminating ambiguous translations is effective as far as we can find alternative translations. However, it is not always the case. An essential approach to solving this problem is to use two or more second languages (Resnik and Yarowsky 2000).

7 Conclusion

Sense-disambiguated association thesauri, in which word senses are distinguished according to the related words, were proposed. It is produced through aligning pairs of related words between association thesauri of different languages. To overcome both the problem of ambiguity in the translingual alignment of pairs of related words and that of disparity of topical coverage between the association thesauri of different languages, an iterative algorithm for calculating the correlation between the senses of a polysemous word and its related words according to the set of words related to both the polysemous word and each of the related words was developed. An experiment using English and Japanese association thesauri, both of which were generated from newspaper article corpora, demonstrated that the algorithm produces a sense-disambiguated association thesaurus successfully. The usefulness of the sense-disambiguated association thesauri for information retrieval and filtering was also discussed.

Acknowledgments

This research was sponsored in part by the Telecommunications Advancement Organization of Japan and the New Energy and Industrial Technology Development Organization of Japan.

References

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1991. Word-sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the ACL*, pages 264-270.

Dagan, Ido and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4): 563-596.

Jing, Yufeng and W. Bruce Croft. 1994. An association thesaurus for information retrieval. In *Proceedings of a Conference on Intelligent Text and Image Handling "RIAO'94"*, pages 146-160.

Kaji, Hiroyuki, Yasutsugu Morimoto, Toshiko Aizono, and Noriyuki Yamasaki. 2000. Corpus-dependent association thesaurus for information retrieval, In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 404-410.

Kikui, Genichiro. 1998. Term-list translation using monolingual word co-occurrence vectors. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 670-674.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the ACL / the 8th Conference of the EACL*, pages 64-71.

Mandala, Rila, Takenobu Tokunaga, and Hozumi Tanaka. 1999. Combining multiple evidence from different types of thesaurus for query expansion. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development of Information Retrieval*, pages 191-197.

Miller, George A. 1990. WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4): 235-312.

Rapp, Reinhard. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 320-322.

Resnik, Philip and David Yarowsky. 2000. Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2): 113-133.

Schuetze, Hinrich and Jan O. Pedersen. 1994. A cooccurrence-based thesaurus and two applications to information retrieval. In *Proceedings of a Conference on Intelligent Text and Image Handling "RIAO'94"*, pages 266-274.

Yarowsky, David. 1993. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 266-271.

Designing Text Filtering Rules: Interaction between General and Specific Lexical Resources

Antonio Balvet
Université Paris X-Nanterre / UMR MoDyCo
200, avenue de la République 92001 Nanterre
antonio.balvet@u-paris10.fr

Abstract

In this paper, we present a modular linguistic wizard for information retrieval applications based on explicit rules. We focus on the main features of the present version of the linguistic wizard: extracting rough verb subcategorization frames from existing corpora and querying a large coverage, corpus-independent semantic network (i.e. Memodata's Dictionnaire Intégral). We also provide performance evaluation measures computed on the basis of a rules-based text filtering system, in order to quantify the gain achieved by making use of the linguistic wizard. The performance evaluation figures are therefore based on a manual run and a "random" run, which provide, respectively, the maximum and minimum quality bounds for a system filtering texts through explicit rules.

1. Introduction

How to provide the right information to the right person at the right time ? This question has become all the more crucial in automatic Information Retrieval (IR) systems, which have to deal with ever-increasing volumes of data. The question at hand, which is, in fact, about relevancy, also applies to the field of Information Filtering (IF). Automatic IF systems, let them be statistics-based or rules-based, are rapidly confronted to the issue of enhancing their initial performance.

In this paper, we show how to integrate both corpus-driven and corpus-independent resources in order to provide more relevant information to the final user.

We first give a historical background of the field of IF, from H.P. Luhn's initial specifications to the current TREC[1] definition. Then, we justify our approach to IF, which is based on explicit categorization rules. In the following section, we present the main features of a LInguistic wiZARD and the gain which can be attained by integrating the LIZARD into the text categorization process, compared to a manual approach.

1.1. From Selective Dissemination of Information to Text Filtering

Providing relevant information is a standard requirement for information systems, let them be human or computer-assisted. This requirement was formally stated in (Luhn, 1958), in the initial framework of public libraries. Luhn was one of the first authors to specify the task which was later to be known as "Information Filtering". The then called "Selective Dissemination of Information" (SDI) activity specified every aspect of a process aimed at fulfilling a full-scale information service, from profiles (information needs) to social filtering (collaborative filtering).

1.2. Filtering Texts: a TREC Definition

The TREC international evaluation conferences, sponsored mainly by the United States' federal government, have taken Luhn's initial specifications to their farthest point, providing the field of Information Retrieval (IR) with standard evaluation procedures as well as standardized tasks and data (gigabytes of text corpora).

1.2.1. Text Filtering as a "Push" Activity

Within the general framework of IR, the IF task was first formalized in 1995. The IF "track", as specified in (Lewis, 1995), is defined as belonging to the range of "push" activities, as opposed to "pull" ones. This means IF is a task where queries (profiles) are stable while the textual data are dynamic (high update rate).

1.2.2. A Binary Selection Decision

The TREC conferences also defined IF as implementing a "binary text classification". The emphasis laid on the binary (YES/NO) aspect of the selection decision distinguishes IF from other push activities such as routing[2], where texts are classified according to a relevance rate computed mainly on the basis of the occurrence probability of a given set of terms (continuous selection decision).

We state that the TREC definition of IF implies an approach to the problem of automatic text classification based on explicit rules, while the routing definition implies a machine-learning, or even statistics-based, one, as explicit rules directly implement binary pattern-matching.

2. Categorizing Text with Rules

2.1. Why Use Rules ?

2.1.1. Explicit *vs.* Implicit Categorization Rules

Machine-learning approaches rely on large amounts of learning material and on the fine-tuning of the often time and space-consuming learning algorithms used. These characteristics make the machine-learning approaches suitable to the classification of stable data repositories, and for activities that do not require -even close to- real-time processing. That is to say that these approaches are particularly well suited to *pull* activities, where data are stable and queries are transient.

[1] Text REtrieval Conference, see (Harman, 1993).

[2] See (Robertson & Hull, 2001) for an overview of the filtering track's subtask (adaptive and batch filtering, routing) specifications.

These approaches are also well adapted to the evaluation procedures defined in the TREC conferences, which are based on a two-stage process[3] for defining reference corpora. The first phase collects all the evaluated systems' outputs, for precedent editions of the evaluation conference[4], from which a portion is extracted, proof-read by human assessors in the second phase[5]. This portion of the original collection is considered as the reference (test) corpus for all evaluated systems.

2.1.2. Real-Scale Data and Explicit Rules

Real-world IF does not fit well in the frame of the TREC conferences, though. As will be seen later in the paper, the available data in actual applications (both "learning" and "testing" corpora) are sometimes quite scarce, amounting to the maximum to megabytes rather than gigabytes of text, thus ruling out *de facto* data-intensive approaches. Furthermore, most of the relevant text units have very low occurrence rates[6], to such extent that detecting these "low signals" appears fundamental to the task of filtering documents. This constitutes yet another indirect justification for the use of symbolic rules, inherently independent from occurrence rates.

2.2. What Rules to Use ?

2.2.1. Keywords-Based Pattern-Matching

In the field of rules-based IF systems, keyword-based pattern-matching approaches are the most common ones. Most of the keyword-based systems are but instances of the renown "grep" command found on Unix-like systems. In keywords-based systems, filters are constituted of search strings, and profiles are Boolean operations on individual filters (NOT, AND, OR). Matching, thus filtering, is limited to exact match of a given string.

2.2.2. Regular Expressions-Based Pattern-Matching

Regular expressions-based IF systems are more flexible than keywords-based ones, in the sense that wildcards (+ and * operators), Boolean (&, |, !) and range (e.g. [a-z]) operators allow for extended search patterns[7]. Those basic features are the building blocks for efficient IF systems. Nevertheless, regular expressions-based IF systems are limited by their syntax, which naïve users are not always willing to master.

[3] See (Voorhees & Harman, 2001) for more details.

[4] This procedure is known as the "pooling method".

[5] The pooling method appears common to all text-related tasks, even the filtering track. Given that most of the evaluated systems rely on implicit categorization rules, this evaluation procedure clearly disfavors alternative approaches, such as explicit rules-based ones.

[6] Named entities (e.g. person/products/company names) register very low occurrence rates compared to other text units; in some cases, non-ambiguous persons/products/companies are only mentioned once.

[7] For example, the following search pattern retrieves all conjugated forms of the French verb "manger": mang*, together with "mangue", "mangeoire" etc..

Neither isolated keywords nor regular expressions appear appropriate for filtering texts: the cost of developing text categorization rules based solely on those basic elements appears overwhelmingly high. Therefore, once stated the necessity of using explicit rules for filtering texts, we need to investigate alternative explicit rules.

3. Local Grammars as Text Filtering Rules

In this section, we introduce corpus-processing oriented symbolic rules: "local grammars" as defined in (Gross, 1975). We show how these local grammars can be used for specific tasks such as text filtering, following the approaches introduced in (Grefenstette, 1996) and (Roche, 1993), who use cascades of Finite State Transducers (FST) for Natural Language Processing-related tasks, in an iterative fashion[8].

3.1. The Local Grammars Approach

Alongside the chomskyan "classical" paradigm for Natural Language Processing (NLP), alternative approaches exist, focusing more on the phrase than on the sentence level, even though pursuing the same goal of arriving at a complete description of human natural language.
Harris's "link grammar"[9] and Gross's "local grammars" are instances of such alternative approaches.

3.1.1. Describing Complex Lexical Units

We focus on the concept of local grammars, such as illustrated in the work of the LAboratoire d'Automatique Documentaire et Linguistique (LADL), and implemented through the Intex platform[10].
Local grammars rely heavily on a distributional analysis of a given corpus. They describe linguistic constituents which are closer to idiomatic phrases than to general sentences, which other distributionalist authors such as B. Habert have named "complex lexical units". Most of the time, local grammars capture very contextual properties of lexical items. Thus, the local grammars approach appears very productive for specialized domains/fields of expertise and terminology-oriented tasks.
The Intex system is based on local grammars, expressed as Finite-State Transducers (FST), which are used as a formalism, as a parsing technique and as a data structure for linguistic knowledge representation. Preprocessing rules (sentence boundaries detection, input normalization), tagging dictionaries (simple and compound words, frozen expressions, named entities etc.) and parsing rules are thus represented as FSTs. This has the effect of ensuring optimal consistency in both data and processes, together with processing efficiency (speed) and extensibility[11]. Moreover, Intex comes with standard large-coverage lexical resources for French: simple and compound words dictionaries, lexicon-

[8] Information processed in earlier stages constrain subsequent analyses. See (Abney, 1996) for other applications such as parsing.

[9] Introduced in (Harris, 1968) and developed ever since.

[10] See (Silberztein, 1993).

[11] Extending/revising a set of local grammars boils down to editing symbolic rules, expressed in a graphical format for better readability (see Figure 1).

grammar tables for frozen expressions, and specialized local grammars (occupational nouns, toponyms, dates, roman numerals etc.).
Figure 1 below shows an example of a very simple local grammar, used to describe, parse and translate roman to modern numerals (transducer output). This very simple local grammar allows for parsing and transformation of input strings: the pattern to match is described in the boxes (e.g. I, II, IX …), the output of the transformation is written in bold (e.g. 1, 2, 9 …).

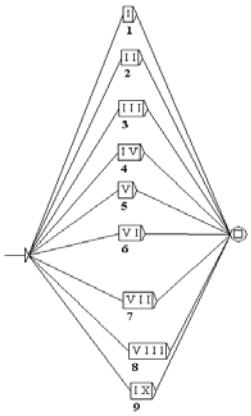


Figure 1: a local grammar used to parse and transform roman numerals into modern numerals

The Intex system also allows for multiple embedding of local grammars, ensuring sufficient computational power for the most common cases by extending FSTs to Augmented Transition Networks.

3.1.2. Describing "Topical Signatures" as local grammars

Our approach to text filtering aims at:

- isolating typical complex lexical units of a given domain/field of expertise, which we call "topical signatures", through a distributional analysis of reference corpora, close to terminological studies in its philosophy,
- describing those expressions as a set of local grammars,
- use this set of local grammars in the process of text categorization.

Typical expressions are thus mainly taken from reference corpora, nevertheless we also make use of thesaurus-like resources in order to provide better coverage for our topical signatures. The approach described here is close to Riloff's[12] in its philosophy, except that topical signatures range from single (e.g. non-ambiguous person names) to complex units (typical phrases such as "monter au capital de"), rather than word pairs exclusively.

3.2. Profiles and Filters as Local Grammars

Filtering textual information involves at least two objects: the user's personal information need, which will be referred to as a "profile", and the individual filters matching relevant parts of documents.
In a rules-based approach, a profile is a conjunction/disjunction or negation of existing filters. In our approach, both filters and profiles can be expressed as local grammars: profiles are conjunctions/disjunctions or negations of existing local grammars matching textual sequences considered relevant by experts of the field.
For example, in order to automatically retrieve relevant documents about the "Mad Cow Disease" epidemics, local grammars for detecting phrases stating the following facts could be designed: typical symptoms have been found on animals, animals have been put down in order to prevent contagion, then perform a Boolean conjunction (AND) operation on those filters in order to implement a "Mad Cow Disease" profile.
Translating filters and profiles into local grammars is consistent with the Intex system's convention. Nevertheless, it implies rendering users' knowledge of the field explicit, which is an inherent source of limitation in coverage of the problem. In some cases, finding categorization rules based on textual cues would even seem awkward, as users rely on implicit, rather than explicit, knowledge and synthetic, rather than analytic, categorization strategies. In those cases, messages are understood in a global fashion and users rely more on their experience of the field than on the actual textual cues contained in the messages. Therefore, the local grammars approach is inherently limited in coverage, even though it complies fully with the TREC specifications[13].

3.3. Problems with Designing Local Grammars by Hand

3.3.1. Experience from a Functional Prototype

A functional prototype of an information filtering system based on local grammars has been designed at a French corporate research laboratory[14]. The prototype, connected to the Agence France Press (AFP) newswire, has demonstrated the feasibility and usability of a rules-based approach to text categorization, together with processing efficiency on French news extracts (ranging from 1 to 10 Kbytes): average processing time (input normalization, filtering and routing) was estimated to 30 seconds per document, which is inferior to the AFP newswire update frequency (1 document per minute).
Nevertheless, the prototype has also shown the necessity to semi-automatically expand user-designed filters, as users cannot explicitly predict future utterances related to a particular domain/area of expertise. In other words, the operational prototype lacked "linguistic calculus" features.

3.3.2. Managing "Flat" Local Grammars

In day-to-day practice, users are quickly confronted to resources management issues due to the proliferation of very specialized (context-dependent) local grammars.

[12] See (Riloff, 1994), where the author presents a strategy focused on extracting non-ambiguous pairs of words from text corpora for "portable" text classification systems.

[13] Our experience of the field has shown us that the TREC specifications for the text filtering task do not account for the complex cognitive (categorization) operations involved in human text filtering.

[14] See (Balvet *et al*., 2001) for more details.

Moreover man-made local grammars are often too restrictive: for example, common phrase alternations (passive/active voice, nominalization etc.) are not available as a standard resource, therefore users usually develop very rough and imperfect grammars for such alternations. Semantic expansion is not implemented in the Intex platform either. Thus, users are rapidly confronted to the problem of expanding their local grammars in a semi-automatic fashion for better coverage and reusability.

4. LIZARD, Main Features

In this section, we introduce the concept of expanded local grammars, and the tools available for French in order to achieve a kind of semi-automatic query expansion on user defined local grammars used as filters, through the LIZARD system.

LIZARD is a tool we have designed, allowing the integration of heterogeneous lexical resources. It was built using the Open Agent Architecture, which provides efficient agent and remote-access capability to heterogeneous systems: OAA allows the creation of Java/C/C++ and Prolog-based agents. The current version of the LInguistic wiZARD is still in alpha status, providing minimal expansion of local grammars: inclusion of synonyms and hyper/hyponyms of terms found in the user's local grammars is suggested, by querying a Memodata agent. Extension to semantically related verbs, together with their preference selection frames extracted from the reference corpora is made available by querying a verb selection preference database.

Syntactic variants are also made available through the following transformations, implemented via local grammars: passive/active form, nominalization with support-verb (e.g. augmenter son capital → procéder à l'augmentation de son capital), and multiple insertions (adjectives, adverbs, phrases etc.).

4.1. Overview of the Global Architecture

4.1.1. A System-Oriented Application

The figure below presents the general system-oriented architecture of the linguistic wizard. Each box in the system diagram represents a processing module. Each module offers standard linguistic and corpus-related facilities, based on existing components, following a "component off-the-shelf" philosophy: every module is thought as a service, therefore each particular component can be replaced by another equivalent component[15].

The Intex module's services are all FST-related text operations (text normalization, pattern-matching, local grammars editing).

The Memodata module's services are all semantics-related operations (retrieving semantically/morphologically related words and phrases, comparing pairs of words or phrases).

[15] For example, the Intex module can be replaced by AT&T's FSM package.

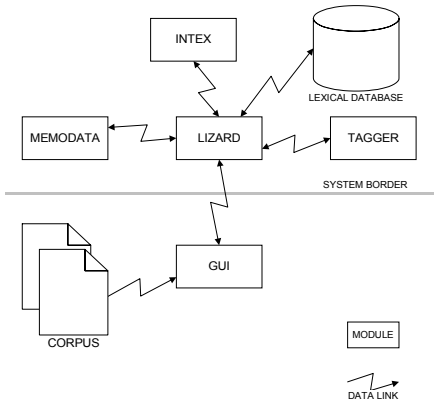


Figure 2: system-oriented architecture of the LIZARD

Communication paths (queries and responses) are represented by broken arrows. The gray line represents the "visible" limits of the whole LIZARD system: the only module accessible by the end-user is the Graphical User Interface (GUI).

The output of the system is a lexical database of domain-dependent typical expressions, which we call "topical signatures".

4.1.2. An Agents-Based System

Developing an agents-based system on top of the modular application shown in Figure 2 was rendered possible by the integration of Stanford Research Institute's Open Agent Architecture (OAA). Within this framework, turning a software component into an autonomous agent is rather straightforward: each module provides services and all agents communicate in a "blackboard" fashion via a central supervising agent called "Supervisor". The Supervisor centralizes all requests from all declared agents and routes them to the appropriate service-rendering agents.

Designing an agent-based NLP system allows the system to operate in a distributed (client/server) fashion over a network (intra/internet), so that memory-intensive applications, such as Memodata's Dictionnaire Intégral, can be run on a dedicated server.

4.2. Rough Verb Subcategorization Frames Extraction

The LIZARD system implements an interactive distributional analysis of reference corpora, in order to extract rough subcategorization frames for relevant verbal entries. For this task, the reference corpora need to be unambiguously tagged and lemmatized, so that only one tag per individual word remains. A first customizable generalization phase deletes most of the Adjectives, all the Adverbs, numbers and punctuation signs. This first phase only keeps those parts of speech generally considered informationally relevant, such as Nouns (part-of-speech information only), Determiners, Verbs (infinitive form), Prepositions and Pronouns.

A second generalization phase provides general subcategorization frames such as: V-Det-N, V-Prep-Det-N etc. Those frames form the core of the domain's set of topical signatures. Once the subcategorization frames have been extracted and validated by the user, all

selected topical signatures candidates are transformed in order to conform to the lexicon-grammar format[16], which the Intex system translates into local grammars[17].

4.3. Querying a Semantic Network

4.3.1. Integrating the Dictionnaire Intégral

Memodata's Dictionnaire Intégral (DI), a corpus-independent semantic network, is presented in detail in (Dutoit, 2000), therefore we only mention the features used by the LIZARD system. The DI comes with a Java API, allowing easy integration in existing systems. This API gives access to common word functions such as synonymy, hyper/hyponymy, morphological relatedness etc. It also gives access to less common features, such as phrase and sentence functions. Those functions are essential to our system, in that they allow easy retrieval of semantically related phrases, not just words. Those functions also allow rough semantic evaluation of two phrases based on a proximity algorithm developed by D. Dutoit.

4.3.2. Expanding Core Topical Signatures

The candidate topical signatures extracted from reference corpora in the previous phase are expanded by querying the DI for related words and phrases: hyper/hyponyms[18], morphologically related words[19] and related phrases are interactively integrated into the existing core topical signatures[20]. The general philosophy is to compensate lack of coverage of hand-designed local grammars by integrating common (extracted from the DI) as well as specific knowledge (extracted from reference corpora) into local grammars intended to be used for automatic text categorization tasks.

5. Performance Evaluation

In the following evaluation, we compare the performance of three text filtering systems[21], following an evaluation procedure aiming at emphasizing the gain attainable by integrating the LIZARD in a rules-based text filtering system.

The first one, the "manual" system, uses hand-designed local grammars[22] and sets the upper bound in quality for the evaluation runs.

The second one, the "computer-assisted" system, is based on the LIZARD and allows us to evaluate our local grammars expansion approach.

The third one, the "random" system, uses random filtering rules and simulates a black-box, automatic text categorization system. This system sets the lower bound for the evaluation runs: we expect our system to perform at least better in quality than the random system.

5.1. The Corpus

5.1.1. A Financial News Corpus

The corpus comes from a private company, Firstinvest, providing targeted financial news to its customers. The financial news extracts are routed by human operators to the appropriate clients in a binary fashion. Thus, the corpus constitutes a reference for an automatic IF system: the situation described matches the TREC definition for the document filtering track.

The reference corpus is organized as follows: 2.6 Mo of French financial news extracts in ASCII format, 19 topics (from Internet-related news to profit warning, rumors and interviews). We focus on topic 19, "corporate transactions", describing scenarios of companies buying or selling parts of their capital.

The performance evaluation measures we used (see below) are based on the number of matches **and** the number of incorrectly retrieved documents (i.e. negative examples) registered for the tested system. As the entire corpus has reached us completely sorted, providing us only with positive examples for each topic, we needed to provide a set of negative examples (noise). Therefore, 50 news extracts (66 Kbytes) of noise corpus, assigned to other topics than the one tested here, were extracted manually from the whole corpus for evaluation purposes.

5.1.2. Learning and Test Corpus

Topic 19 totals 303 documents, which we segment in two parts: 2/3 for the learning corpus (200 documents) and 1/3 (103 documents) for the test corpus. For each evaluation run, standard precision and recall rates (P/R, see below) were computed based on the comparison between each system's output and the reference corpus from Firstinvest.

As the reader will undoubtedly notice, these figures are very far from those of evaluation conferences such as TREC, even though they correspond to real-life data. In fact, the reference data we describe can not be compared to the reference corpora provided by TREC editions: the documents were sorted entirely by hand, they represent but a fraction, in size, of the TREC test suites, and they match an actual information need from users ready to pay for the service provided by Firstinvest.

5.2. Setting the Upper and Lower Bounds to Evaluate the LIZARD Approach

5.2.1. The Manual Run

S. Bizouard designed a set of local grammars for an information extraction (IE) system evaluation experiment undertaken at Thales RT. Following E. Riloff, we assert that IF and IE are complementary activities. Thus, IE local grammars can be used as IF profiles. Therefore, we took S. Bizouard's hand-designed local grammars as a reference for the manual run. Those

---

[16] Syntactic and semantic information, associated to a lexical entry, are expressed as a set of binary features (+/-). Lexicon-grammar tables also include lexical parameters such as the form of a typical complement.

[17] See (Silberztein, 1999) for more details on the lexicon-grammar feature of the Intex system.

[18] Specifics and generics in Memodata's terminology.

[19] For example: "achat" (Noun) which is morphologically related to "acheter" (Verb).

[20] The current version of the LIZARD does not make use of the semantic net navigation customization features, implemented in the DI, yet.

[21] Performing a form of "batch" filtering according to the definition of (Robertson & Hull, 2001).

[22] See (Bizouard, 2001) for more details.

resources were designed following the topical signature approach described above.
Precision and recall of S. Bizouard's grammars do not equal the theoretical 100%, even though they are the result of considerable effort[23]. Our hypothesis is that this apparent lack of coverage is mainly due to implicit knowledge used by experts of the field in classifying texts, which explicit approaches such as the one described in this paper can not capture. The apparent lack of coverage of the hand-designed local grammars also appears due to a lack of proper selection preference constraining: some rules remain too "open" by failing to provide a closed list of possible complements for some very common verbs[24].
Our implicit hypothesis is that manually-designed resources tend to rate high in precision, but low on recall, so the manual run will give the higher precision bound.

5.2.2. The LIZARD Run

The computer-assisted run shows the impact of the integration of both corpus-driven and corpus-independent resources on a text categorization task. In other words, the computer-assisted run implements a query expansion approach based on explicit resources (verb subcategorization frames, semantically related words ...).
The implicit hypothesis is that the natural low recall rates tendency of the manual approach can be compensated by elements (parameters) taken both from existing specialized corpora and general purpose semantic nets (i.e. Memodata's Dictionnaire Intégral).

5.2.3. The "Random" Run

This run is based on a fully automatic text filtering system, which randomly selects documents, independently from their content. The random run shows what can be achieved by a text filtering system which decision selection rules are hidden (black-box system). The implicit hypothesis is that the random run will set the lower bound for both recall and precision (around 50%), in other words, the minimal recall and precision rates expected from the computer-assisted system.

5.3. Figures

RUN	Matches	Noise
Manual	76	9
LIZARD	103	13
Random[25]	53.2	24.8

Figure 3: performance table for each run

These figures were computed on the test (103 documents) and noise corpus (50 documents). As the table shows, the LIZARD system retrieves all the relevant documents. Moreover, it only is responsible for about 1/3 additional noise (compared to the manual run).
The figures presented below give the standard precision/recall rates for each run[26]. As the figures show, the LIZARD system performs very good in recall (100%) and compares equally to the manual run in precision, despite of its "noise" rate being slightly higher.

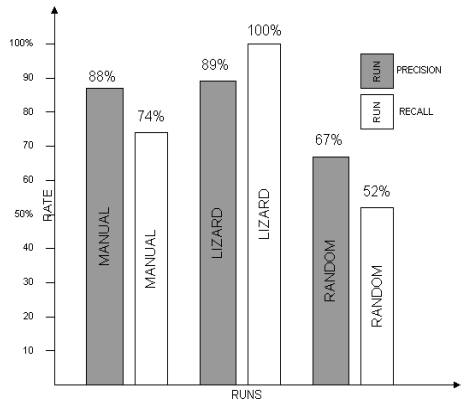


Figure 4: precision/recall rates of three text filtering approaches

5.3.1. Discussion of Figures

The figures presented above show the performance of three types of text filtering systems:

- a system relying exclusively on manually-designed categorization rules, centered on topical signatures,
- a system based on computer-assisted categorization rules (topical signatures), integrating mainly subcategorization frames extracted from the learning corpus, and suggestions from a thesaurus agent,
- a system relying on unknown categorization rules, which appear to be random.

The figures appear consistent with the implicit hypotheses: the "manual" system rates high in precision (88%) but rather low in recall (74%). The manual run validates our "topical signatures" approach, it also shows that explicit approaches fail at capturing part of the knowledge used by experts in a text categorization task.
The "random" system rates moderately in recall (around 50%: 52% in average over 10 runs) and rates rather well in precision (67%). This would appear surprising, should one not bear in mind the essential property of random processes, together with the binary nature of the selection decision evaluated here. In other words, faced with 2 possibilities (select/discard), the random system performs exactly as expected, as it would have for a coin-flipping output prediction simulation: it gives around 50% correct answers[27]. Still, the "rules" used in the decision selection process can not be traced back,

[23] Approximately 3 man-months.
[24] E.g. complements for the verb "céder" are not specified, while it can be found in phrases such as "céder sa filiale" but also in "céder à ses avances" which is not related to topic19.
[25] The rates for the random system were averaged over 10 runs, given the random nature of the system tested.
[26] Precision = Nb. of matches / Nb. of responses,
Recall = Nb. of matches / Total of expected responses.
[27] Respectively, incorrect.

while tracing and debugging capabilities are inherent to symbolic approaches. In other words, the "random" system would appear to perform surprisingly well in regard to its cost[28] if not for its opaque way of categorizing text, its fickle selection decision[29] and its "black box" nature. The random system also shows the relative efficiency of our approach: in the classical evaluation framework described, relying on external evidence (recall and precision rates), almost 50% of the problem are covered without any "intelligence" whatsoever.

Finally, the figures computed for the LIZARD run show the substantial gain attainable by integrating both common and specific knowledge in the text categorization process. The LIZARD approach thus provides the field of information filtering with a seemingly viable and efficient approach, even though complementary experiments should take place in order to evaluate more precisely the gain of the local grammars expansion approach.

5.4. Conclusion and Perspectives

In this paper, we have shown how the field of Information Retrieval, i.e. Information Filtering, could benefit from a symbolic approach to text classification tasks such as "batch filtering". Moreover, we have shown that real-life data, consisting of a corpus of short specialized texts (financial news), did not fit well in the frame of the international TREC evaluation conferences, providing gigabytes of textual data and evaluation procedures that favor data-intensive (machine-learning) approaches. Therefore, in order to evaluate the approach described, we have presented a procedure which compares our system's performance to a manual and a random one, rather than figures based on the official "utility" measures for text filtering systems' evaluation.

We have tried to show how the integration of hybrid resources - corpus-driven (specialized) and corpus independent (general) ones - in the design process of automatic categorization rules expressed as Finite-State Transducers could yield better results than rules designed solely by hand. The figures presented show the performance of LIZARD, a system based on interactively expanded symbolic rules for automatic text filtering, which rates high in recall and compares equally well in precision to a manual approach.

The experiments described in this paper have also shown us that even though human operators' expertise is crucial to the IF activity, it is not less prone to subjectivity than other categorization tasks. Therefore, any attempt to compare the performance of a given IF system to a human reference should take into consideration the problem of the inherent subjectivity attached to the IF/categorization task. In other words, we plan to follow qualitative (glass-box) evaluation procedures in the future, rather than purely quantitative (black-box) ones.

[28] Easy implementation, low space/memory load.

[29] The retrieved document set varies with every run.

6. References

Abney, S. (1996). Partial Parsing via Finite-State Cascades. *Proceedings of the ESSLLI'96 Robust Parsing Workshop*.

Balvet, A. Meunier, F. Poibeau, T. Viard, D. Vichot, F. Wolinski, F. (2001). Filtrage de Documents et Grammaires Locales : le Projet CORAIL. *Actes du congrès de l'ISKO (International Society for Knowledge Organisation)*, 5-6 juillet 2001. Université de Nanterre-Paris X.

Bizouard, S. (2001). *Évaluation d'Outils d'Acquisition de Ressources Linguistiques pour l'Extraction*. Mémoire de DESS en Ingénierie Multilingue. CRIM, INALCO.

Dutoit D., (2000). *Quelques Opérations Texte → Sens et Sens → Texte Utilisant une Sémantique Linguistique Universaliste Apriorique*. Ph.D. dissertation. Caen University.

Grefenstette, G. (1996). Light Parsing as Finite-State Filtering. *Workshop on Extended Finite State Models of Language, ECAI'96*.

Gross, M. (1975). *Méthodes en Syntaxe*. Paris, Hermann.

Harman, D. (1993). Overview of the First Text REtrieval Conference (TREC-1). *NIST Special Publications*. Gaithersburg, MD.

Harris, Z.S. (1968). *Mathematical Structures of Language*. Interscience Publishers, John Wiley & Sons.

Lewis D., Hill M. (1995). The TREC-4 filtering track. *NIST Special Publications*. Gaithersburg, MD.

Luhn, H.P. (1958). A Business Intelligence System. *IBM Journal of Research and Development*, Vol 2(4), pp. 314-319.

Riloff, E. (1994). *Information Extraction as a Basis for Portable Text Classification Systems*. Ph.D. dissertation. University of Massachussets Amherst.

Robertson, S. & Hull, D.A. (2001). The TREC-9 Filtering Track Final Report. *NIST Special Publications*. Gaithersburg, MD.

Roche, E. (1993). *Analyse Syntaxique Transformationnelle du Français par Transducteurs et Lexique-Grammaire*. Ph.D. dissertation. Paris VII University.

Silberztein, M. (1993). *Le Système INTEX, Dictionnaires Électroniques et Analyse Automatique des Textes*. Paris, Masson.

Silberztein, M. (1999). Traitement des Expressions Figées avec INTEX. *Linguisticae Investigationes*, tome XXII, pp. 425-449. John Benjamins Publishing Company.

Voorhees, E. & Harman, D. (2001). Overview of the Ninth Text REtrieval Conference (TREC-9). *NIST Special Publications*. Gaithersburg, MD.

Lexically-Based Terminology Structuring: a Feasibility Study

Natalia Grabar, Pierre Zweigenbaum

DIAM — STIM/DSI, Assistance Publique – Hôpitaux de Paris
& Département de Biomathématiques, Université Paris 6
{ngr,pz}@biomath.jussieu.fr

Abstract

Terminology structuring has been the subject of much work in the context of terms extracted from corpora: given a set of terms, obtained from an existing resource or extracted from a corpus, identifying hierarchical (or other types of) relations between these terms. The present work aims at assessing the feasibility of such structuring by studying it on an existing, hierarchically structured terminology. For the evaluation of the results, we measure recall and precision metrics, taking two different views on the task: relation recovery and term placement. Our overall goal is to test various structuring methods proposed in the literature and to check how they fare on this task. The specific goal in the present phase of our work, which we report here, is focussed on lexical methods that match terms on the basis on their content words, taking morphological variants into account. We describe experiments performed on the French version of the US National Library of Medicine MeSH thesaurus. This method proposes correct term placement for up to 26% of the MeSH concepts, and its precision can reach 58%.

1. Background

Terminology structuring, *i.e.*, organizing a set of terms through semantic relations, is one of the difficult issues that have to be addressed when building terminological resources. These relations include subsumption or hyperonymy (the *is-a* relation), meronymy (*part-of* and its variants), as well as other, diverse relations, sometimes called 'transversal' (*e.g.*, *cause*, or the general *see also*).

Various methods have been proposed to discover relations between terms (see (Jacquemin and Bourigault, 2002) for a review). We divide them into *internal* and *external* methods, in the same way as (McDonald, 1993) for proper names. Internal methods look at the constituency of terms, and compare terms based on the words they contain. Term matching can rely directly on raw word forms (Bodenreider et al., 2001), on morphological variants (Jacquemin and Tzoukermann, 1999), on syntactic structure (Bourigault, 1994; Jacquemin and Tzoukermann, 1999) or on semantic variants (synonyms, hyperonyms, etc.) (Hamon et al., 1998). External methods take advantage of the context in which terms occur: they examine the behavior of terms in corpora. Distributional methods group terms that occur in similar contexts (Grefenstette, 1994). The detection of appropriate syntactic patterns of cooccurrence is another method to uncover relations between terms in corpora (Hearst, 1992; Séguéla and Aussenac, 1999).

The present work aims at assessing the feasibility of such structuring by studying it on an existing, hierarchically structured terminology. Ignoring this existing structure and starting from the set of its terms, we attempt to discover hierarchical term to term links and compare them with the preexisting relations.

Our aim consists in testing various structuring methods proposed in the literature and checking how they fare on this task. The specific goal in the present phase of our work, which we report here, is focussed on lexical methods that match terms on the basis on their content words, taking morphological variants into account.

After the presentation of the data we used in our experiments, we present methods for generating hierarchical links between terms through the study of lexical inclusion and for evaluating their quality with appropriate recall and precision metrics. We then detail and discuss the results obtained in this evaluation.

2. Material

In this experiment we used an existing hierarchically structured thesaurus, a 'stop word' list, and morphological knowledge.

2.1. The MeSH biomedical thesaurus

The Medical Subject Headings (MeSH, MeS (2001)) is one of the main international medical terminologies (see, *e.g.*, Cimino (1996) for a presentation of medical terminologies).

It is a thesaurus specifically designed for information retrieval in the biomedical domain. It is used to index the international biomedical literature in the Medline bibliographic database. The French version of the MeSH (INS, 2000) contains a translation of these terms (19,638 terms) plus synonyms. It happens to be written in unaccented, uppercase letters.

As many other medical terminologies, the MeSH has a hierarchical structure: 'narrower' concepts (children) are related to 'broader' concepts (parents). The MeSH specifically displays a rich, polyhierarchical structure: each concept may have several parents. In total, the MeSH contains 26,094 direct child-to-parent links and (under transitive closure) 95,815 direct or indirect child-to-ancestor links.

2.2. Stop word list

The aim of using a 'stop word' list is to remove from term comparison very frequent words which are considered not to be content-bearing, hence 'non-significant' for terminology structuring.

The stop word list used in this experiment is a short one (15 word forms). It contains the few grammatical words which occur frequently in MeSH terms, articles and prepositions:

au, aux, d', de, des, du, en, et, l', la, le, les, ses, un, une

2.3. Morphological knowledge

Previous work has acknowledged morphology as an important area of medical language processing and medical information indexing (Pacak et al., 1980; Wingert et al., 1989; Grabar et al., 2002) and of term variant extraction (Jacquemin and Tzoukermann, 1999). In this work, we apply morphological knowledge to the terminology structuring task.

Three types of morphological relations are classically considered:

- *Inflection* produces the various forms of a same word such as plural, feminine or the multiple forms of a verb according to person, tense, etc.: *intervention – interventions*, *acid – acids*. The parts of speech of a lemma and its inflected forms are the same. Reducing an inflected form to its lemma is called lemmatization.

- *Derivation* is used to obtain, *e.g.*, the adjectival form of a noun (noun *aorta* ↔ adjective *aortic*, verb *intervene* ↔ noun *intervention*, adjective*human* ↔ adverb *humanely*). Derivation often deals with words of different parts of speech. Reducing a derived word to its base word is called stemming.

- *Compounding* combines several radicals, here often of greek or latin origin, to obtain complex words (*e.g.*, *aorta* + *coronary* yields *aortocoronary*).

The morphological knowledge we used consists of {*lemma*, *derived or inflected form*} pairs of word forms where the first is the 'normalized' form and the second a 'variant' form. The general principle is that both forms of such a pair have similar meaning.

In this work we rely on inflectional knowledge and derivations that do not change word meaning. We have left compounding aside for the time being, since the words it relates may have distant meanings.

2.3.1. Inflectional knowledge

For inflection, we have two lexicons of such word pairs. The first one is based on a general lexicon (ABU, `abu.cnam.fr/DICO`) which we have augmented with pairs obtained from medical corpora processed through a tagger/lemmatizer (in cardiology, hematology, intensive care, and drug monographs): it totals 219,759 pairs (where the inflected form is different from the lemma). The second lexicon is the result of applying rules acquired in previous work (Zweigenbaum et al., 2001) from two other medical terminologies (ICD-10 and SNOMED) to the vocabulary in the MeSH, ICD-10 and SNOMED (total: 2,889 pairs).

2.3.2. Derivational knowledge

For derivation, we also used resources from (Zweigenbaum et al., 2001) which, once combined with inflection pairs, result in 4,517 pairs.

These morphological resources will still need to be improved; but we believe that the results should not vary much from what is present here.

3. Methods

The present work induces hierarchical relations between terms when the constituent words of one term lexically include those of the second term (section 3.1.). We evaluate these relations by comparing them with the pre-existing relations, computing precision and recall both for links and concepts (section 3.2.).

3.1. Lexical Inclusion

The method we use here for inducing of hierarchical relations between terms is basically a test of *lexical inclusion*: we check whether a term P (*parent*) is 'included' in another term C (*child*). We assume that this type of inclusion is a clue of a hierarchical relation between terms, as in the following example: *acides gras* / *acides gras indispensables* (*fatty acids* / *fatty acids, essential*).

To detect this type of relation, we test whether all the content words of P occur in C. We test this on segmented terms with a gradually increasing normalization on word forms:

- basic normalization: conversion to lower case, removal of punctuation, of numbers and of 'stop words' (introduced in section 2.2.);

- normalization with morphological ressources (see section 2.3.): lemmatization (with the two alternative inflectional lexicons) and stemming with a derivational lexicon.

Terms are indexed by their words to speed up the computation of term inclusion over all term pairs of the whole MeSH thesaurus. When these normalizations are applied, terms are indexed by their normalized words: we assume that P is lexically included in C iff all normalized words in P occur in C.

3.2. Evaluation

We evaluated the results obtained with this approach by comparing them with the original structure in the MeSH. We considered two methods to evaluate this terminology structuring task:

- the first method is interested in the number of links found, and compares these links with those originally present in the MeSH thesaurus: do we obtain all the links that pre-exist in the MeSH?

- the second method considers the positioning of individual MeSH concepts (terms) in the hierarchical structure of the thesaurus: can we place each concept in at least one suitable position in the emerging hierarchy?

For both methods, we compute recall and precision metrics. The recall metric allows us to analyze the completeness of the results and to know whether all the expected links are induced and concepts positioned. The precision metric evaluates the correctness of induced results.

The recall and precision measures computed here have two versions:

- strict (only the links to direct parents of a given concept are considered satisfactory), and
- tolerant (a link to any ancestor is considered as correct).

We also tested a mixed scheme: the weight given to each link depends on the distance between the two concepts related with this link in the original hierarchical structure of the MeSH: the more distant these concepts, the lower the weight the induced link obtains. However, since the mixed scheme results are not very different from the tolerant one, we do not present them here.

The lexical inclusion methods and the evaluation procedure were implemented as Perl5 scripts.

4. Results

4.1. Lexical inclusions obtained

The method described in section 3.1. has been applied to the flat list of 19,638 terms ('main headings') of the MeSH thesaurus. The gradualy increasing normalizations we applied to this list of terms allow us to induce an increasing number of hierarchical links between these terms.

In table 1 we show quantitative results for the relations induced with the analysis of lexical inclusions and obtained with each type of morphological normalization tested. The first column introduces the types of normalization. The *raw* results were obtained with no morphological normalization. The *lem-gen* results were obtained with application of inflection pairs compiled from a general lexicon, and *lem-med* results with inflectional pairs acquired from medical terminologies (see section 2.3.1.). The *lem-stem-med* results correspond to the normalization done with derivational pairs (see section 2.3.2.). The basic normalization (conversion to lower case, removal of punctuation, numbers and stop words) is performed in all cases. The second column presents the number of links induced with each of the normalization methods tested. The third column recalls the number of hierarchical relations in the MeSH.

Type of normalization	Number of links	Reference
raw	9,189	95,815
lem-gen	12,963	95,815
lem-med	11,627	95,815
lem-stem-med	15,942	95,815

Table 1: Quantification of induced relations berween analyzed terms.

In table 2 we present the same type of information for the placement of terms. The second column contains the number of terms which have been linked with our methods. This number corresponds to the number of concepts that can be linked in the 'structured' terminology we induced. The third column recalls the number of linked terms in the MeSH hierarchy.

As expected, the number of links induced between terms increases when applying inflectional normalization and even more with derivational normalization. Inflectional knowledge compiled from the general lexicon (*lem-gen*) allows to link more terms than that only obtained from specialized terminologies (*lem-med*): 12,963 *vs* 11,627 links. But for the positionining of terms, we obtain better covering of terms when using specialized morphological knowledge (*lem-med*) than when using morphological knowledge from general lexicon (*lem-gen*): 10,949 *vs* 10,261 terms.

Type of normalization	Number of terms	Reference
raw	9,126	19,638
lem-gen	10,261	19,638
lem-med	10,949	19,638
lem-stem-med	11,752	19,638

Table 2: Quantification of positioned terms.

Lemmatization can be ambiguous when an inflected form can be obtained from several lemmas (*e.g.*, *souris* $\rightarrow$ *souris/N* (*mouse*) and *sourire/V* (*to smile*)). In that case, we have adopted a brute force approach which merges the two corresponding morphological families and chooses one lemma as unique representative for both.

Table 3 shows examples of lexically included terms which we obtained with this method. For each type of normalization, it shown pairs *parent* / *child* corresponding to direct, then indirect relations in the original MeSH structure.

4.2. Evaluation of these lexical inclusions

In section 3.2. we presented the methods designed to evaluate the structuring results we obtain with a lexical inclusion analysis of terms. These methods allow us to evaluate recall and precision metrics for both relations between terms and term positioning. In all the cases we take into account the nature of induced links (direct or indirect ones) by testing both strict and tolerant variants. The correctness of induced results is computed by comparing these results with the original MeSH structure.

Table 4 shows the evaluation results for the links, and table 5 for concept (term) placement.

The second column in table 4 contains the number of direct and indirect correct links; the third column shows the number of incorrect links (links which do not exist in the MeSH). The *Recall, direct* column presents the recall R_d of the direct links found d (weighted by the number of direct links $D = 26,094$ in the MeSH – see section 2.1.); the *Recall, all* column presents the recall R_a of all the links (weighted by the total number of links $D + I = 95,815$ in the MeSH):

$$R_d = \frac{d}{D}; R_a = \frac{d+i}{D+I}$$

The last column of this table presents the evaluation of the precision metric, taking into account both strict and tolerant appoaches; if d is the number of direct links found, i the number of indirect links found, and n the number of non-MeSH links found, strict precision P_s and tolerant precision P_t are:

$$P_s = \frac{d}{d+i+n}; P_t = \frac{d+i}{d+i+n}$$

Type of normalization	Parent P	Child C
raw $_{direct}$	accouchement	accouchement provoque
	delivery	*labor, induced*
raw $_{indirect}$	acides gras	acides gras indispensables
	fatty acids	*fatty acids, essential*
lem-gen $_{direct}$	intervention chirurgicale	interventions chirurgicales obstetricales
	surgical procedures, operative	*obstetric surgical procedures*
lem-gen $_{indirect}$	intervention chirurgicale	interventions chirurgicales voies biliaires
	surgical procedures, operative	*biliary tract surgical procedures*
lem-med $_{direct}$	agents adrenergiques	inhibiteurs captage agent adrenergique
	adrenergic agents	*adrenergic uptake inhibitors*
lem-med $_{indirect}$	chromosomes humains	chromosome humain 21
	chromosomes, human	*chromosomes, human, pair 21*
lem-stem-med $_{direct}$	aberration chromosomique, anomalies	aberrations chromosomes sexuels, anomalies
	chromosome abnormalities	*sex chromosome abnormalities*
lem-stem-med $_{indirect}$	eosinophilie	poumon eosinophile
	eosinophilia	*pulmonary eosinophilia*

Table 3: Examples of correct, lexically induced MeSH terms, and their English translations. Indirect means that the MeSH includes a path of length > 1 from the parent to the child.

Normalization	Correct links		Incorrect	Recall (%)		Precision (%)	
	direct	indirect	(non MeSH)	direct	all	strict	tolerant
raw	2688	1266	5235	10.3	4.1	29.3	43.0
lem-gen	3058	1779	6790	11.7	5.0	26.3	41.6
lem-med	3451	2171	7341	13.2	5.9	26.6	43.4
lem-stem-med	3580	2316	10046	13.7	6.2	22.5	37.0

Table 4: Recall and precision of lexically-induced links.

Normalization	Recall: correct advices / # MeSH nodes			Precision: correct advices / # advices		
	strict (%)	tolerant (%)	MeSH nodes	strict (%)	tolerant (%)	nodes linked
raw	10	18	19543	27	52	6969
lem-gen	10	23	19543	24	55	8078
lem-med	10	26	19543	24	58	8644
lem-stem-med	9	26	19543	18	55	9398

Table 5: Recall and precision of lexically-induced node placement advices.

The recall of links increases when applying more complete morphological knowledge (inflection then derivation). And, not surprisingly, we notice that the recall of relations between terms obtained with morphological knowledge acquired from medical terminologies (*lem-med*, *lem-stem-med*) is higher (13.2 and 13.7%) than the recall corresponding to the use of the morphological knowledge compiled from the general lexicon (*lem-gen*, 11.7%).

The evolution of precision is opposite: injection of more extensive morphological knowledge (derivation *vs* inflection) leads to taking more 'risks' for generating links between terms: *raw* results precision is 29.3% *vs* 22.5% for *lem-stem-med* precision.

When accepting both direct and indirect links (tolerant approach), the precision measure obtained is higher than when only direct links are considered (strict approach). For instance, with raw normalization, the tolerant approach gives a precision of 43.0% and the strict approach 29.3%. For the *lem-stem-med* normalization the tolerant precision is 37.0% and the strict precision is 22.5%.

Depending on the normalization and the weighting scheme, up to 29.3% of the links found are correct, and up to 13.7% of the direct MeSH links are found by lexical inclusion.

Up to 26% of the concepts are correctly placed under their ancestors; and the term positioning advices are correct in up to 58% of the cases.

5. Discussion

We presented in this paper an experiment of terminology structuring. We tested here some 'internal' methods for this task, which consist in the analysis of the lexical inclusions of terms. We consider that a term P is lexically included in a term C iff all words of P occur in C, and that this is a clue of its being a parent (ancestor) of C. To help this analysis we apply normalizations, both basic and making use of morphological knowledge.

Whereas raw lexical inclusion detects directly attainable relations between terms by matching identical words in these terms, lemmatization adds flexibility with inflectional variants. Morphological stemming allows to link terms which contain words that are graphically different but have a very close meaning. This allows to obtain hierarchical dependencies between terms that are more based on the 'meanings' of these terms. These semantic similarities are detected through the morphological analysis we apply.

To assess the induced results we compare them with the original structure of the MeSH. We evaluate both the induced links and the placed terms. Depending on the normalization and the weighting scheme, up to 29.3% of the links found are correct, and up to 13.7% of the direct MeSH links are found by lexical inclusion. Up to 26% of the terms are correctly placed under their ancestors; and the placement advices are correct in up to 58% of the cases.

The only expected and evaluated type of relation is the hierarchical one, as exists in the MeSH thesaurus. But we assume that the methods applied also allow to induce other types of relations, and maybe other hierarchical relations, which are not in the original MeSH hierarchy. Some 'new' relations can be found, for instance, in the incorrect ('extra'-) relations we induced. These additional relations have to be analysed in a detailed way to better evaluate the results obtained with these simple methods.

In summary, lexical inclusion caters for a non-negligible number of the hierarchical concept organization in the MeSH thesaurus; and the use of morphological knowledge, mainly for lemmatization, significantly increases this proportion. As could have been hypothesized, trying to place a concept at one position in the hierarchy is more successful than finding all the links from this concept to its parents in a polyhierarchical terminology.

A simple analysis of lexical inclusions shows that in many cases a hierarchical dependency between (medical) terms can be detected and allows to obtain an important number of hierarchical relations between these terms. This information is useful when dealing with the terminology structuring task.

To detect and evaluate more relations between terms, other methods for terminology structuring may be applied, such as those presented in section 1. We plan to test them in the same context as the morphological experiments presented here.

6. References

Olivier Bodenreider, Anita Burgun, and Thomas C. Rindflesch. 2001. Lexically-suggested hyponymic relations among medical terms and their representation in the UMLS. In URI INIST CNRS, editor, *TIA'2001 Terminologie et Intelligence artificielle*, pages 11–21, Nancy.

Didier Bourigault. 1994. Extraction et structuration automatiques de terminologie pour l'aide à l'acquisition de connaissances à partir de textes. In *RFIA'94*, pages 1123–1132. AFCET.

James J Cimino. 1996. Coding systems in health care. In Jan H. van Bemmel and Alexa T. McCray, editors, *Yearbook of Medical Informatics '95 — The Computer-based Patient Record*, pages 71–85. Schattauer, Stuttgart.

Natalia Grabar, Pierre Zweigenbaum, Lina Soualmia, and Stéfan J. Darmoni. 2002. A study of the adequacy of user and indexing vocabularies in natural language queries to a MeSH-indexed health gateway. *J Am Med Inform Assoc*, 8(suppl). *Submitted*.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Natural Language Processing and Machine Translation. Kluwer Academic Publishers, London.

Thierry Hamon, Adeline Nazarenko, and Cécile Gros. 1998. A step towards the detection of semantic variants of terms in technical documents. In Christian Boitet, editor, *Proceedings of the 17 th COLING*, pages 498–504, Montréal, Canada, 10–14 August.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In Antonio Zampolli, editor, *Proc 14 th COLING*, pages 539–545, Nantes, France, 23–28 July.

Institut National de la Santé et de la Recherche Médicale, Paris, 2000. *Thésaurus Biomédical Français/Anglais*.

Christian Jacquemin and Didier Bourigault. 2002. Term extraction and automatic indexing. In Ruslan Mitkov, editor, *Handbook of Computational Linguistics*. Oxford University Press, Oxford. *To appear*.

Christian Jacquemin and Évelyne Tzoukermann. 1999. NLP for term variant extraction: A synergy of morphology, lexicon, and syntax. In Tomek Strzalkowski, editor, *Natural language information retrieval*, chapter 2, pages 25–74. Kluwer Academic Publishers, Dordrecht & Boston.

David D. McDonald. 1993. Internal and external evidence in the identification and semantic categorization of proper names. In Branimir Boguraev and James Pustejovsky, editors, *Corpus Processing for Lexical Acquisition*, pages 61–76. MIT Press, Cambridge, MA.

2001. Medical Subject Headings. WWW page `http://www.nlm.nih.gov/mesh/meshhome.html`, National Library of Medicine, Bethesda, Maryland.

M. G. Pacak, L. M. Norton, and G. S. Dunham. 1980. Morphosemantic analysis of -ITIS forms in medical language. *Methods Inf Med*, 19:99–105.

Patrick Séguéla and Nathalie Aussenac. 1999. Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine. In Régine Teulier, editor, *Actes de IC'99*, June.

F. Wingert, David Rothwell, and Roger A Côté. 1989. Automated indexing into SNOMED and ICD. In Jean Raoul Scherrer, Roger A. Côté, and Salah H. Mandil, editors, *Computerised Natural Medical Language Processing for Knowledge Engineering*, pages 201–239. North-Holland, Amsterdam.

Pierre Zweigenbaum, Stéfan J. Darmoni, and Natalia Grabar. 2001. The contribution of morphological knowledge to French MeSH mapping for information retrieval. *J Am Med Inform Assoc*, 8(suppl):796–800.

Query Expansion by a Contextual Use of Classes of Nouns

Gaël de Chalendar and Brigitte Grau

LIR group – LIMSI (CNRS)
BP 133 91403 Orsay Cedex
Gael.de.Chalendar@limsi.fr, Brigitte.Grau@limsi.fr

Abstract

We developed a system, SVETLAN', dedicated to the acquisition of classes of semantically close nouns from texts. We aim at constructing a structured lexicon for the general language, that is not for representing a specialized domain. Thus, texts are open-domain newspaper articles. The acquisition is based on a distributional method that groups the nouns that are related to a same verb with a same functional role. However, in order to deal with polysemy, classes are learned in context: they are built from text segments related to a same semantic domain. For that, we use results of ROSA, a system that clusters automatically segmented texts in order to build semantic domain defined by sets of weighted words. We will show how these classes can be used to expand queries, in comparison with an expansion realized by using WordNet.

1. Introduction

Information Retrieval systems often require semantic knowledge to improve their results. However, one can ask, "what type of semantics?". According to the application, it may differ. It can be only synonymous, or semantically close words, or words belonging to a same domain, either specific or general. One conclusion is that it is necessary to be able to bring together words with close signification. Moreover this gathering has to be done in a well defined context in order to take into account multiple meanings of words. For example, in the context of nuclear plants, one confronted to the sentences: *"… started to replace the fuel rods…"*, *"… started to replace the combustible of the reactor…"* and *"… to replace the films and the batteries of the camera… "*, should join together the words *combustible* and *rods* but should put aside the word *film*.

We are interested in robust applications aimed to cope with every domain, opposed to domain specialized systems. Those systems often use preexistent knowledge to find synonyms or related words but it remains difficult to select the right information. For instance, the noun *care* has 6 registered meanings in WordNet 1.6 (Fellbaum, 1998). If we are interested in medicine practice, we do not want to retrieve documents that use the word *care* with its 4th sense ("*a cause to feeling concerned*"), but maybe only those that use it with its first sense: "*the work of caring for […] someone […]*".

Our conclusion after these statements is that a general ontology or classification seeking for universality is an utopia and principally because of the word polysemy. So, the terminological aspect of general language has to be modeled by multiple overlapping classifications. The question we have to ask is then: "how can these classifications be acquired". We make three hypotheses. Firstly, at least a part of the semantic knowledge is encoded in the texts. Secondly, a part of this text-encoded knowledge can be automatically extracted and lastly, this extraction will be feasible only if semantics is considered in fine-grained contexts.

Work has been done during previous decades on general language but the encoding was mainly manual, as for scripts of Schank (Schank, 1982) that were defined for storing semantico-pragmatic representations of everyday situations. It has been proved very difficult to extend the scripts beyond the first few ones. Another example of manually encoded semantic knowledge is CYC (Lenat, 1986) that is supposed to be a universal semantic knowledge base. In reality, CYC has to be manually tuned in each application it is used in.

On the contrary, various methods have been used with success to acquire semantic knowledge on specialized domains: cooccurrences statistics (Zernik, 1991), distributional approaches following Harris ideas (Harris, 1968), classification techniques (Agarwal, 1995), linguistic indices (Roark & Charniak, 1998), etc. Our interrogation was on the possibility of adapting these successful techniques to general language. Our proposition is to determine automatically thematic domains and to apply a classical distributional method on texts belonging to a same domain. This approach allows our system to form classes of semantically close words.

The idea behind the distributional method is that the usage of a verb is directed by its sub-categorization frame. This frame specifies for example that the subject of the verb should be an instance of a particular concept. The set of real objects referred to by the words that are subjects of the verb in a particular domain represent this concept by extension. Thus, a description of this extension is the set of words used to refer to these objects. These sets of words are the semantic classes made by our system, SVETLAN' (Chalendar & Grau, 2000).

We will show how these classes can be used to expand queries, in comparison with an expansion realized by using WordNet.

2. Overview of the system

Input data of SVETLAN' (see Fig. 1) are semantic domains with the thematic units (TUs) that have given birth to them. Domains are sets of weighted words, relevant to represent a same specific topic. These domains are automatically learned by ROSA that aggregates similar thematic units, made of sets of words. TUs are built by a topic segmentation process relying on lexical cohesion. It processes texts such as newspaper articles.

The first step of SVETLAN' consists of a syntactic parsing of the corpus in order to produce the structured thematic units (STUs) corresponding to each TU. STUs are constituted by a set of triplets - a verb, the head noun of a phrase and its syntactic role - extracted from the parser results. The STUs related to a same semantic domain are aggregated altogether to learn a structured domain. Aggregation leads to group nouns playing the same syntactic role with a verb in order to form classes.

As these aggregations are made within STUs belonging to a same domain, classes are context sensitive, which ensures a better homogeneity. A filtering step, based on the weights of the words in their domain allows the system to eliminate nouns from classes when they are not very relevant in this context.

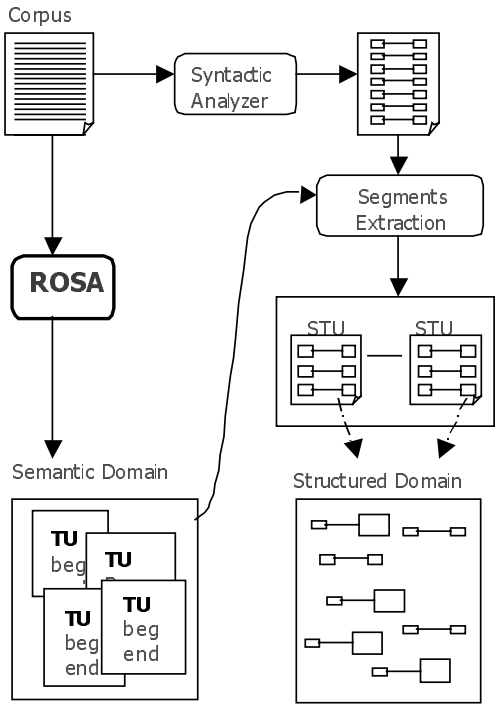


Figure 1: Schemata of Structured Domain learning

3. The ROSA system

We only give here a brief overview of the system that is made of two modules, SEGCOHLEX and SEGAPSITH. It is described more precisely in (Ferret & Grau, 1998). ROSA incrementally builds topic representations, made of weighted words, from discourse segments delimited by SEGCOHLEX (Ferret, 1998). It works without any *a priori* classification or hand-coded pieces of knowledge. Processed texts are typically newspaper articles coming from the *Los Angeles Times*. They are pre-processed to only keep their lemmatized content words (adjectives, single or compound nouns and verbs).

The topic segmentation implemented by SEGCOHLEX is based on a large collocation network, built from 24 months of the *Los Angeles Times* newspaper, where a link between two words aims at capturing semantic and pragmatic relations between them. The strength of such a link is evaluated by the mutual information between its two words. The segmentation process relies on these links for computing a cohesion value for each position of a text. It assumes that a discourse segment is a part of text whose words refer to the same topic, that is, words are strongly linked to each other in the collocation network and yield a high cohesion value. On the contrary, low cohesion values indicate topic shifts. After delimiting segments by an automatic analysis of the cohesion graph, only highly cohesive segments, named thematic units (TUs), are kept to learn topic representations. This segmentation method entails a text to be decomposed in small thematic units, whose size is equivalent to a paragraph. Because discourse segments, even related to the same topic, often develop different points of view of this topic, we enrich the particular description given by a text. We add to the TUs those words of the collocation network that are particularly linked to the words found in the corresponding segment.

Words	occ.	weight
examining judge	58	0.501
police custody	50	0.442
public property	46	0.428
charging	49	0.421
to imprison	45	0.417
court of criminal appeal	47	0.412
receiving stolen goods	42	0.397
to presume	45	0.382
criminal investigation department	42	0.381
fraud	42	0.381

Table 1: The most representative words of a domain about justice

Learning a complete description of a topic consists of merging all successive points of view, i.e. similar TUs, into a single memorized thematic unit, called a semantic domain. Each aggregation of a new TU increases the system's knowledge about one topic by reinforcing recurrent words and adding new ones. Weights on words represent the importance of each word relative to the topic and are computed from the number of occurrences of these words in the TUs (see Table 1 for an example of a domain). This method, implemented in SEGAPSITH, leads to learn specific topic representations as opposed to (Lin, 1997) for example whose method builds general topic descriptions as for economy, sport, etc.

4. Semantic Domain Structuring

Semantic domains are similar to classes formed by (Zernik, 1991). SVETLAN' purpose is then to delimit small classes inside these domains, and to associate them to the verbs they define, as it is made in distributional approaches (Faure & Nedellec, 1998) (Pereira & al., 1993). A class is defined by those nouns which play a same role relative to a same verb and that are supposed to be connected by a strong semantic link. Thus, even if they do not denote a same object, the objects denoted by them play a similar role in the tight context defined by the semantic domain.

4.1. Formation of The Structured Thematic Units

A syntactic parser processes texts in order to find the verbs and their arguments. For English, we used the link grammar (Grinberg & al., 1995). The system extracts all the triplets found by the analyzer, constituted by a verb, a syntactic relation and the head noun of the noun phrase. Relations are subject, direct and indirect objects, the preposition that introduces a prepositional phrase. The link grammar only gives one interpretation of the sentence.

After parsing the texts, SVETLAN' groups the triplets relatively to the delimited thematic units. So, we define a structured thematic unit as a set of <*Verb→syntactic*

relation→Noun> structures, i.e. a syntactic relation instantiated with a verb and a noun. We will refer to these structures as instantiated syntactic relations.

4.2. Aggregation

Structured thematic units related to a same domain are aggregated altogether to form the structured domains. Aggregating a structured thematic unit within a structured domain consists of:

- aggregating the instantiated syntactic relations that contain the same relation and the same verb, i.e. associating a set of words to an argument of a verb;
- adding new instantiated syntactic relations, i.e. adding new verbs with their arguments made of a syntactic relation and the lemmatized form of a noun.

Nouns are not weighted inside a class; they only keep the weight they had in their semantic domain. Thus, the criterion to define a class is that words appear with a same verb, in similar contexts. The similarity of contexts is a lexical similarity computed on the whole domain.

5. Results

Classes are built according to two levels of contextual use of the words: a global similarity of the thematic contexts and a local relevance inside a domain we added to discard irrelevant words. In order to illustrate the effect of topic similarity when building classes, we show in Table 2 a class regrouping all the direct objects found for the verb *to replace* in the whole corpus. We can see that there is no semantic proximity between those nouns. When the class is formed, for the same verb, inside a nuclear domain, the class is then homogeneous. So, even general verbs, as *to replace* (it is possible to replace a lot of things), are relevant criteria to group nouns when their appear in similar thematic units.

to replace	*object*	text, constitution, trousers, combustible, law, dinar, rod, film, circulation, judge, season, device, parliament, battalion, police, president, treaty
to replace	*object*	combustible, rod

Table 2: The effect of the thematic context on the kind of classes

However, classes of nouns contain a lot of words that disturb their homogeneity. These words often belong to parts of the different TUs at the origin of the semantic domain that are not very related to the described topic. They correspond to meanings of words scarcely used in the current context. As these words are weekly weighted in the corresponding domains, the data can be filtered: each noun that possesses a weight lower than a threshold is removed from the class. By this selection, we reinforce learning classes of words according to their contextual use.

to establish	*object*	base, zone
to answer	*to*	document, question, list
~~**to establish**~~	~~*object*~~	~~base, zone~~
to answer	*to*	document, question, ~~list~~

Table 3: Two filtered classes in a domain about nuclear weapons

Table 3 shows two aggregated links obtained without filtering in its upper part and the filtered counterparts in its lower part. The link for the verb '*to establish*' has been completely removed while the link of the verb '*to answer*' with the preposition 'to' has been reduced by the removing of '*list*'.

Table 4 shows some examples of classes obtained by SVETLAN'. Even when verbs are polysemous, which is the case for several verbs in the examples, the domain membership constraint leads the system to build relevant classes. We also can see that the various syntactic relations are relevant criteria to gather semantically linked words.

Domain	Verb	Relation	Class
War	to qualify	Direct Object	president, leader
Food assistance	to take refuge	Into	country, region
Tour de France	to cover	Direct Object	stage, tour
Sport	to face	In	match, final
Economy	to release	Direct Object	million, billion
Festival cinema	to tell	Subject	film-maker, film
Conflict Croatia	to resume	Direct Object	negotiation, discussion
Economy	to reduce	Direct Object	surplus, deficit

Table 4: Examples of noun classes

SVETLAN' originality relies on the constitution of classes given with their context of reference. As a context is explicitly defined by a set of words, it gives indices, when finding a word in a text or a sentence, to choose a class or another, and so to obtain neighbor words. We will show the application of this property when expanding a query.

Verb	Relation	Class
To accuse	Subject	Indictment, prosecutor
	By	Prosecutor, jury
To make	Subject	Prosecutor, indictment
	Direct Object	Jury, prosecutor
To show	Subject	Juror, defendant
	Direct Object	Jury, scheme
To tell	Subject	Magistrate, informant
	Direct Object	Juror, jury
To give	Direct Object	Sentence, prosecutor, trial
	From	Sentence, prosecution
	To	Jury, defendant

Table 5: Example of verbs with classes defining their arguments in a domain about justice

However, the constitution of classes is not the sole result of SVETLAN'. The structuring of semantic domains is another. Instead of bag of words, domains are

now described by verbs associated to classes defining their arguments. This kind of knowledge is a first step towards schema representation of pragmatic knowledge. Such an example is given in Table 5.

6. Experiments

6.1. Corpus Characteristics

We conducted an experiment with a corpus of English newspaper articles composed of 3 months of the "Los Angeles Times" newspaper. We used the following experimental settings: segmentation of the corpus and creation of the thematic memory (i.e. the set of semantic domains); syntactic analysis and syntactic links extraction; structured memory creation (i.e. the set of structured domains); and lastly, an evaluation of the results. We first counted the number of correct classes. A correct class is one that contains words sharing a direct semantic link. For the wrong classes, we counted the number of errors due to parse errors.

For our experiment, we only keep the TUs that lead to build stable domains, i.e. domains grouping at least 10 TUs.

The corpus we worked on is unanalyzed and SGML encoded. Its language level is high with a journalistic style and it tackles various topics. The size of corpus is 7.3 million words.

6.2. Results

The thematic memory created contains 138 stable domains. Table 8 shows results obtained with these domains. Within about 150 classes, about 60% are correct while 7% of wrong classes are due to parse errors.

Number	Correct	Syntactic Parser Errors	Other
149	58 %	7 %	35 %

Table 8: Results on English with a 0.1 threshold

Table 9 shows some examples of the classes contained in a structured domain whose topic is medicine.

Verb	Rel^{on}	Class
To take	Under	Home, residence
To meet	Object	Care, physician
To carry	Object	Virus, antibody
To get	Subject	Treatment, care

Table 9: Examples of classes in a structured domain on English

These examples show two classes with the word care. They instantiate two different kinds of semantic relation: in the class <care, treatment> we see an instrument link between the two terms of the class (a treatment is a means to take care of a patient) and in the class <care, physician>, the link is an agent one (the physician take care of his patients). Meanwhile, in the same structured domain, there were other classes containing the word *care*, some of them carrying the same meaning as care considered as a treatment. So classes do not partition the words of the domains, and they also do not partition the meanings of the words. In a further step, we will study if it is possible and suitable to merge the closest classes.

7. Query Expansion

We were interested in knowing which effects are produced by using different sorts of knowledge in query expansion. Thus, we did some preliminary experiments. Given a query made of words, we tried two kinds of expansion. One kind exploited the acquired classes and the other WordNet. WordNet is a lexical database made by lexicographers. It aims at representing the sense of the bigger part of the lexicon. It is composed of Synsets. A Synset is a set of words that are synonymous. These Synsets are linked by *IS-A* relations. We only did few experiments whose purpose was only to illustrate the interest of having contextual classes compared to a general database which often creates divergences when used as it is.

First, we selected the domain the closest to the query words. Different expansions where computed by adding the words that were belonging to the class of a word of the initial query, and this for each word of the query belonging to a class in the selected domain.

By this way, expansion is done relatively to the query domain of reference. It should be noted that another expansion might be done from a same word from another query, as soon as the other words of the query differ and refer to another context. On the contrary, when expanding with WordNet, the lack of domain knowledge does not allow to select only the right sense.

The queries were sent on Google, that only considers the first 10 words. We chose Google because it is a boolean engine, assuming that when the query contain a lot of words, the retrieved documents are more relevant, as they contain all the words of the query. It is also a way of showing the validity of the acquired classes. If there exists documents containing all the words of the expanded query, the class can be considered coherent. So, in this experiment, we tried to shorten the initial set of documents retrieved by Google.

Initial query : *prosecutor obstruction deliberation jury*
=> 477 documents
SVETLAN' query expansion 1: prosecutor obstruction deliberation jury charge case court trial attorney count
=> 141 answers
SVETLAN' query expansion 2: prosecutor obstruction deliberation jury charge case court trial attorney sentence
=> 222 answers

When using WordNet, we retrieved the different meanings of each word – first, all its synonyms and its hypernyms and second, only the synonyms – and add each of these sets to the initial query. Such a set was considered equivalent to an acquired class. Thus for the same initial query, we obtained the following query expansions.

1 sense of prosecutor (its synonymous and after "=>" its hypernyms)
Sense 1: prosecutor, prosecuting officer, prosecuting attorney
=> lawyer, attorney

Initial query : *prosecutor obstruction deliberation jury*
WordNet expansion 1: prosecutor obstruction deliberation jury, prosecuting officer, prosecuting attorney, lawyer, attorney
=> 65 answers

4 senses of obstruction
Sense 1 :obstruction, impediment, impedimenta
=> structure, construction
Sense 2: obstacle, obstruction
=> hindrance, deterrent, impediment, handicap
Sense 3: obstruction
=> hindrance, interference, interfering
Sense 4: obstruction
=> maneuver, manoeuvre, play

Initial query : *prosecutor obstruction deliberation jury*
WordNet expansion 2: prosecutor obstruction deliberation jury, impediment, impedimenta, structure, construction, obstacle, hindrance, deterrent, handicap, interference, interfering
=> No answer
WordNet expansion 2bis: prosecutor obstruction deliberation jury, impediment, impedimenta, obstacle
=> No answer

5 senses of deliberation
Sense 1: deliberation
=> discussion, give-and-take, word
Sense 2: deliberation, weighing, advisement
=> consideration
Sense 3: calculation, deliberation
=> planning, preparation, provision
Sense 4: slowness, deliberation, deliberateness, unhurriedness
=> pace, rate
Sense 5: deliberation, deliberateness
=> thoughtfulness

Initial query : *prosecutor obstruction deliberation jury*
WordNet expansion 3: prosecutor obstruction deliberation jury, discussion, give-and-take, word, weighing, advisement, consideration, calculation, planning, preparation, provision, slowness, deliberateness, unhurriedness, thoughtfulness
=> No answer
WordNet expansion 3bis: prosecutor obstruction deliberation jury, weighing, advisement, calculation, slowness, deliberateness, unhurriedness
=> No answer

2 senses of jury
Sense 1: jury
=> body
Sense 2: jury, panel
=> committee, commission

Initial query : *prosecutor obstruction deliberation jury*
WordNet expansion 4: prosecutor obstruction deliberation jury, discussion, body, committee, commission
=> 84 answers

We can see that expansions along the WordNet synonyms of polysemous words do not lead to a successful research, as for *deliberation* and *obstruction*. An explanation of this result comes from the fact that SVETLAN's added words are much more related to the query than those added via WordNet. It is due to the contextual construction of the classes and also to the fact that the context is explicitly represented by domains and so can be used to guide the choice of words, contrarily to what happen when using WordNet. WordNet coverage is large but this quality is, in a sense, its shortcoming. Indeed, the generality of its contents makes it difficult to use in real sized applications. It rarely can be used without a lot of manual adaptation.

We are now showing another example, in the sport domain. SVETLAN' added words that all belong to the baseball domain and also lead to reduce the number of retrieved documents.

Initial query: starter hitter batter : 14900 answers
Svetlan'A expansion: starter hitter batter run hit game inning pitch season home
=> 7660 answers

In WordNet, starter and batter are very polysemous words.

5 senses of starter
Sense 1: starter
=> electric motor
Sense 2: starter
=> contestant
Sense 3: starter, dispatcher
=> official
Sense 4: newcomer, fledgling, fledgeling, starter, neophyte, freshman, entrant
=> novice, beginner, tyro, tiro, initiate
Sense 5: crank, starter
=> hand tool

1 sense of hitter
Sense 1: batter, hitter, slugger, batsman
=> ballplayer, baseball player

2 senses of batter
Sense 1: batter, hitter, slugger, batsman
=> ballplayer, baseball player
Sense 2: batter
=> concoction, mixture, intermixture

In such a case, it is not possible to obtain a correct expansion by using only WordNet.

However, one can envisage using SVETLAN' knowledge to select a meaning in WordNet. By combining on one hand sets of semantic closed words, without explicit types of link, and on the other hand sets of words with typed semantic relations that often are no more semantically closed if they are all merged, we could

maybe use the first sets to select contextual meanings in the second sets.

8. Related Works

There is a lot of works dedicated to the formation of classes of words. These classes have very various statuses. They can contain words belonging to the same semantic field or near synonymous, for example.

Automatic systems apply different criteria to group words, but all make use of a context notion or a proximity measure. IMToolset, by Uri Zernik (Zernik, 1991), cluster local contexts of a studied word that is defined by the 10 words surrounding it in the texts. The proximity between words is evaluated by using the mutual information measure, as we do when segmenting the text. The result is groups of words that are similar to our domains but more focused on the sense of a word alone.

Faure and Nedellec (Faure & Nedellec, 1998) with Asium, or Lin (Lin, 1998) apply distributional approaches to learn classes. Asium was designed to build ontology of specialized domains, so there is no need for a context restriction. Its basic classes are clustered to create ontology by the mean of a cooperative learning algorithm. This manual cooperative part is a step analogous to our filtering step. Lin does not apply a contextual selection of the words before regrouping them; he defined a similarity measure between words of a same class to order them according to their similarity degree, This kind of method also lead to build large classes, analogous to our semantic domains.

9. Conclusion and Future Work

The system SVETLAN' we propose, in conjunction with SEGAPSITH and a syntactic parser, extracts classes of words from raw texts and structures domains initially made of bags of words. These classes are created by the gathering of nouns appearing with the same syntactic role after the same verb inside a context. This context is made by the aggregation of text segments referring to similar subjects. Our experiments on different corpus give good enough results, but they also confirm that a great volume of data is necessary in order to extract a large quantity of lexical knowledge by the analysis of syntactic distributions.

In order to show the interest of building small classes inside larger domains, we made some query expansions that comfort the feeling of real proximity between words in the classes and their interest for specializing a query. We are now studying how this expansion can be used in a question-answering system (Ferret et al., 2001) developed in the group that participated to the TREC evaluations. This task is open domain and when the answer is not expressed in the documents with the same words as the question, it requires finding exact synonyms in text sentences. A first step will consist of augmenting our base by applying our system on much more texts, then trying to use WordNet in conjunction with SVETLAN': a synonym in WordNet would be selected if it occurs in a class of SVETLAN' or in classes very close each others. As SVETLAN' classes do not only contain synonyms, the classes are not sufficient in this case, while used along with WordNet it would be a very sure criterion to obtain synonyms in a specified context. We have to verify that it will be applicable on a large scale.

10. References

R. Agarwal, Semantic feature extraction from technical texts with limited human intervention, PhD thesis, Mississippi State University, 1995.

Gaël de Chalendar and Brigitte Grau, SVETLAN' or how to Classify Words using their Context, Proceedings of 12th EKAW, Juan-les-Pins, France, October 2000, pages 203-216 Rose Dieng and Olivier Corby (Eds.), Springer, 2000, (Lectures notes in computer science; Vol. 1937 : Lectures notes in artificial intelligence).

David Faure and Claire Nedellec, ASIUM, Learning subcategorization frames and restrictions of selection. In Y. Kodratoff ed., proceedings of 10th ECML – Workshop on text mining, 1998.

Christiane Fellbaum, WordNet: an electronic lexical database, The MIT Press, 1998.

Olivier Ferret, How to thematically segment texts by using lexical cohesion? Proceedings of ACL-COLING'98 (student session), pp. 1481-1483, Montreal, Canada, 1998.

Olivier Ferret and Brigitte Grau, A Thematic Segmentation Procedure for Extracting Semantic Domains from Texts, Proceedings of ECAI'98, Brighton, 1998.

O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, C. Jacquemin, Terminological variants for document selection and Question/Answer matching, Proceedings of the Open-Domain Question Answering Workshop, Conference of ACL/EACL, Toulouse, 2001

Dennis Grinberg, John Lafferty and Daniel Sleator, A robust parsing algorithm for link grammars, Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and Proceedings of the Fourth International Workshop on Parsing Technologies, Prague, September, 1995.

Zellig Harris, Mathematical Structures of Language, Wiley, New York, 1968.

C.-Y. Lin, Robust Automated Topic Identification, Doctoral Dissertation, University of Southern California, 1997.

Douglas Lenat, Cyc: a large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11), 1995.

Dekang Lin. Automatic retrieval and clustering of similar words. In Proceedings of COLINGACL '98, pages 768--774, Montreal, Canada, August 1998.

Fernando Pereira, Naftali Tishby and Lillian Lee, Distributional clustering of english words, Proceedings of ACL'93, 1993.

B. Roark and E. Charniak, Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. Proceedings of COLING-ACL'98, pp. 1110-1116, 1998.

Roger C. Schank, Dynamic memory. A theory of reminding and learning in computers and people. Cambridge University Press, 1982.

Uri Zernik, TRAIN1 vs. TRAIN2: Tagging Word Senses in Corpus, RIAO'91, 1991