Inducing Sense-Discriminating Context Patterns from Sense-Tagged Corpora

Anna Rumshisky*, **James Pustejovsky***

*Dept. of Computer Science
Brandeis University
Waltham, MA 02454, USA
{arum, jamesp}@cs.brandeis.edu

Abstract

Traditionally, context features used in word sense disambiguation are based on collocation statistics and use only minimal syntactic and semantic information. Corpus Pattern Analysis is a technique for producing knowledge-rich context features that capture sense distinctions. It involves (1) identifying sense-carrying context patterns and (2) using the derived context features to discriminate between the unseen instances. Both stages require manual seeding. In this paper, we show how to automate inducing sense-discriminating context features from a sense-tagged corpus.

1. Introduction

Pustejovsky et al. (2004) introduced the notion of corpus patterns as knowledge-rich collections of context features that allow humans to disambiguate between different senses of a polysemous word. The Corpus Pattern Analysis technique (CPA), as outlined in Pustejovsky and Hanks (2001), initially requires a human subject to identify the collection of context features needed to disambiguate a particular predicate. The recorded features are further used to disambiguate the unseen instances. In this paper, we present a strategy for automation of an important step in this process, that is, the identification of features relevant for disambiguation.

The idea that semantic similarity between words must be reflected in the similarity of habitual contexts in which words occur is fairly obvious and has been formulated in many guises (e.g. "distributional hypothesis" (Harris, 1985), "strong contextual hypothesis" (Miller and Charles, 1991)). When applied to the case of lexical ambiguity, it translates into looking at the context for disambiguation clues, since one expects that it would be even more true for similar meanings of the same word. They would occur in similar contexts.

In contemporary work on word sense disambiguation, this general notion is used rather uniformly. What varies widely is the representation of the context. Until fairly recently, the typical context representations overwhelmingly used cooccurrence-based features. Each feature corresponded to the frequency with which other words and/or small n-grams occurred within a small window of the target word. Local features typically used a smaller window, topical features could track keywords occurring within a sentence or a paragraph.

In the last few years, context representations used in WSD have increasingly incorporated some syntactic and semantic information. In the recent SENSEVAL-3, for example, several of the better-performing systems that competed in the English Lexical Sample task incorporated syntactic as well as semantic information (Mihalcea et al., 2004). Lee et al. (2004) included features derived from grammatical relations over lemmas and POS tags (parent headword and POS, voice of parent VP, etc.). Agirre and Martinez (2004) tracked WordNet Domains for each context of the target word, as well as several syntactic dependencies (subject, object, noun-modifier, preposition and sibling relations).

Typically, in the evaluation of WSD systems, performance is averaged over all target words. But in reality, how well a particular feature set performs for a given word very much depends on the type of ambiguity involved. The costly human analysis involved in the initial stages of CPA creates an inventory of possibilities for such feature sets. In practice, extracting the features that would capture the fine distinctions available to the human annotator involves applying a variety of preprocessing tools which in many cases produce error. A system that hopes to combine different knowledge sources needs to (1) select the right feature subset for a particular target word and (2) weed out the features that are excessively noisy. In this paper, we report WSD experiments with information gain-based feature selection performed on CPA patterns of several polysemous English verbs.

The rest of the paper is structured as follows. Section 2 reviews the CPA technique and its purpose and direction. Section 3 describes a generic feature extraction architecture we use. Section 4 presents the results of WSD experiments with polysemous verbs.

2. CPA Patterns

CPA is a corpus analysis technique that provides insight into the types of context parameters that allow humans to disambiguate between different predicate senses. The CPA approach refines and extends the scope of typical knowledge-rich context features to include the following elements:

- shallow semantic typing of predicate arguments
- minor syntactic categories (locatives, adjuncts, etc.)
- predicate arguments represented by lexical sets
- subphrasal syntactic cues: genitives, partitives, bare plural/determiner, infinitivals, negatives, collocational cues

Below are selected CPA patterns for the verb "fire". There is typically a many-to-one relation between the patterns and the senses they represent. The distribution of frequencies associated with each sense are typically far from even. The "fire" patterns representing senses that account for more than 5% of use are not listed below. [1]

Selected CPA Patterns for FIRE:

```
I DISCHARGE A GUN AT A TARGET  (60%)

1. [[Person]] fire [[LEXSET Firearm]] (at [[PhysObj]])

2. [[Person]] fire [[LEXSET Projectile]] (off)
   ({from [[LEXSET Firearm]]}) ({at [[PhysObj]]}
   | [ADV[Direction]])

3. [[Person]] fire [NO OBJ] ({at [[PhysObj]]}
   | {on [[HumanGroup]]} | [ADV[Direction]])

4. [[LEXSET Firearm]] fire [NO OBJ] ({at [[PhysObj]]}
   | {on [[HumanGroup]]} | [Adv[Direction]])

III DISMISS AN EMPLOYEE (11%)

6. [[Person 1]] fire [[Person 2]] (for [[Action=Bad]])

VII INSPIRE SOMEONE (11%)

12. [[TopType]] fire {[[Person]]'s [[LEXSET Enthusiasm]]}

13. [[TopType]] fire [[Person]] (up)
```

Phrasal verbs are analyzed separately. Below are selected CPA patterns for the phrasal verb "take off".

Selected CPA Patterns for TAKE OFF:

```
TAKE OFF

24. [[Person]] take [[Garment]] {off}

25. [[Person 1]] take {[COREF POSDET] hat} {off}
    {[PREP to] [[Person 2]]}

26. [[Event]] take {the_smile} {off [[Person]]'s [Face]]}

27. [[Person 1 | Event]] take {weight}
    {off [[Person 2]]'s {shoulders | mind}}

28. [[Person]] take {weight} {off [COREF POSDET] feet}

29. [[TopType]] take {[[Person]]'s mind}
    {off [[TopType = Topic]]}

30. [[Vehicle = Airplane]] take [NO OBJ] {off}
    (for [[Location]])

31. [[Animate]] take [NO OBJ] {off}
    ([PREP to] [[Location]])

32. [[Vehicle]] take [[Person]] {off}
    {[PREP to] [[Location]]}

33. [[Person]] take [REFL-PRON] {off}
    {[PREP to] [[Location]]}

34. [[Process] | [Institution]] take [NO OBJ] {off}

35. [[Person]] take {off} [[Abstract = Quantity]]

36. [[Person]] take [[TimePeriod]] {off}

37. [[Person 1]] take [[Person 2]] {off} {at [[Location]]}

38. [[Person 1]] take [[Person 2]] {off [[Activity]]}

39. [[Person 1]] take [[Person 2]] {off [[Document]]}
```

3. Feature Selection

Recognizing automatically the context patterns that contribute to resolving the ambiguities of each predicate is a serious challenge. Identifying many of the contributing factors can really only be approximated with state-of-the-art preprocessing resources. Therefore, the general problem of feature selection involves being able to (1) combine multiple knowledge sources in feature representations and (2) weed out the noisy features that either propagate error or do not contribute to disambiguation. For the experiments below, we implemented a test system for feature extraction and feature set optimization.

3.1. Generic Feature Extractor

The generic feature extractor extracts grammatical and semantic features from the surrounding context of the target predicate. Currently, the feature extractor uses the following information sources:

(i) RASP Parser (Briscoe and Carroll, 2002)
(ii) Brandeis Shallow Ontology (BSO Lite, Pustejovsky et al. (2004), (2006), (Rumshisky et al., 2006)),
(iii) Context heuristics, information on event senses from WordNet, etc.

For every grammatical relation the target predicate participates in, the following kinds of features are extracted:

(1) stem-populated grammatical relations
(2) POS-populated grammatical relations
(3) grammatical relations populated with BSO semantic types
(4) grammatical relations populated heuristic tags

Heuristic tags presently considered include Plural, Animate, Capitalized, WordNet Event, and aggregate POS tags (nominals, reflexive pronouns, etc.). For example, consider the following two occurrences of the verb "fire" from the BNC:

(1) (a) "..when police fired on the demonstrators with rubber bullets and live ammunition, killing at least seven people."
(b) "The new classical macroeconomists are committed believers in the power of market forces, being fired with an almost evangelistic enthusiasm."

In both instances, RASP extracts the indirect object relation, which is then populated with POS and heuristic and semantic types described above. The following binary features are then added to the feature set. In (1a):

```
(iobj, with/IW, fire/VVN, bullet/NN2)

  full GR:           iobj with/IW fire/VVN bullet/NN2

  stem-only GR:      iobj with fire bullet

  POS-only GR:       iobj IW fire/VVN NN2

  semantic type GRs: iobj with fire TypeMaterialEntity
                     iobj with fire TypePLURAL
```

In (1b):

```
(iobj, with/IW, fire/VVN, enthusiasm/NN1)

  full GR:           iobj with/IW fire/VVN enthusiasm/NN1
```

[1] See (Pustejovsky et al., 2004) for pattern syntax.

```
stem-only GR:       iobj with fire enthusiasm

POS-only GR:        iobj IW fire/VVN NN1

semantic type GRs:  iobj with fire TypeState
                    iobj with fire TypeEvent
```

The feature extractor is designed to allow for easy integration of features deriving from separate data processing pipelines. The training and test data is stored in full sentences, with the answer key for the target attribute of each sentence (corresponding the particular Wordnet sense or CPA pattern number) stored separately. Occurrences of more than one target context within one sentence are ignored at present.

3.2. Feature Set Optimization

Once the set of generic features is extracted for a given predicate stem, the feature selection algorithm uses the training data to discard the features that are likely to have little impact on disambiguating between different senses of the predicate.

The feature set that derives from the training data as described above will obviously include a lot of spurious features that will have little impact on actual disambiguation. Prior to attempting any of the popular dimensionality reduction techniques, we would like to be able to weed out the noise. In order to do that, we need to be able to evaluate how good the feature set presently extracted for a given predicate is, i.e. how well it performs. We evaluate how well a given feature set performs by computing the *feature set precision*, i.e. the precision it gives on a WSD task when used in a machine learning algorithm. Since semantically tagged training data tends to be scarce, we would compute the precision by a variation of held-out cross-validation.

The initial feature set extracted for each predicate is obviously quite large for even the small training sets (over 3000 features for under 400 examples), so computing the feature set precision for each subset of features extracted for a given predicate is not computationally feasible. A straightforward alternative to deciding which subsets of features give better precision would be to successively eliminate each feature from the set, computing the feature set precision on the rest, then eliminating the features with negative impact (or positive impact below a chosen threshold). One of the systems presented at Senseval-3 (Escudero et al., 2004) that employed a similar feature selection procedure used feature addition/deletion in order to optimize the feature set, but the procedure had to be abbreviated due to "computational overhead".

Here, we use a simple information gain-based variation on the above feature selection idea. Predicate sense is the target attribute. The features are sorted on the information gain achieved due to each feature on the full training set, preferring the features that minimize the weighted sum of target attribute entropies after the split. The features with information gain below a certain cutoff are filtered out.

The features that do well on the training set should essentially self-select. Take, for example, grammatical features carrying semantic type information. The Brandeis Semantic Ontology assigns multiple types to lexical items, both due to true multiple typing and to type inheritance. Consider the type that in some argument position actually does contribute to the disambiguation of the target predicate. The feature that carries that type would be extracted for more training instances than the feature carrying a type from a lower level in the semantic hierarchy. On the other hand, the type at a higher level would be too generic to disambiguate between different senses, and thus would be filtered out during the feature weeding.

Unavoidably, some spurious features still end up in the optimized feature set. For example, here is a typical spurious feature that would be retained for "fire" under the filter of information gain < 0.02:

```
(aux, _/ , fire/VVN, have/VHZ)
  information gain:     0.029
```

This feature marks contexts in which "fire" occurs in the present perfect. There are 14 CPA patterns, i.e. fine-grained senses associated with "fire". This feature would be retained under the informaion gain threshold of 0.02, even though it induces a fairly even distribution of target value frequency for the 14 patterns. For example, here are the sense frequency distributions for patterns 1 through 14 for for the two cases: (1) when the spurious feature was detected in the context, (2) when it was absent from context:

- with the spurious feature firing:
  $[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0]$
- with the spurious feature absent from context:
  $[27, 61, 57, 12, 8, 35, 2, 5, 4, 2, 19, 10, 9, 1]$

4. WSD Experiments

Seven verbs with different degrees of polysemy were used in the WSD experiments: "back", "fire", "force", "grasp", "seek", "settle", and "backfire". We computed the feature set precision before and after the removal of low impact features, and then compared both against the baseline, looking at the achieved improvement, if any. The disambiguation experiments here were conducted with the information gain cutoff of 0.05. We use the majority sense as the baseline. Precision testing is done by a variant of held-out cross-validation. In several trials, the sense-tagged data is presplit, with a certain percentage randomly selected as test data. In these experiments, the test set comprised 25% of the sense-tagged data. A decision tree disambiguation algorithm is run on the test set, using (1) the original feature set and (2) the modified feature set. Precision and % improvement over baseline is computed for both. The results of decision tree-based disambiguation are averaged over 5 trials.

4.1. Results

The number of CPA patterns for each verb, along with average precision and improvement over baseline achieved with original (unmodified) and with modified (filtered) feature set is shown in Table 1. Although averaging system performance over semantically diverse ambiguity types is not ideal for evaluation, it is commonly used to compare WSD systems. In our experiments, average per-instance polysemy over all seven verbs combined is 14.7 and the

				Original feature set			Modified feature set		
Verb	No. of senses	sense-tagged instances	avg. baseline	avg. prec.	improvement over baseline	t-test	avg. prec.	impr. over baseline	t-test
back	25	415	26.0%	58.0%	124.7%	$p < .001$	55.4%	117.7%	$p < .001$
fire	14	341	23.7%	27.0%	15.0%	-	35.1%	56.2%	$p < .005$
force	11	332	74.8%	75.4%	0.6%	-	77.1%	3.2%	-
grasp	8	243	62.5%	70.5%	16.3%	$p < .02$	72.1%	12.7%	-
seek	8	229	63.3%	79.2%	27.9%	$p < .005$	71.5	11.0%	$p < .04$
settle	19	327	26.5%	50.5%	98.6%	$p < .001$	54.9%	108.5%	$p < .001$
backfire	2	78	97.5%	96.0%	0.0%	-	99.0%	0.0%	-

Table 1: WSD results for selected polysemous verbs. Number of senses reflects the number of CPA patterns for the verb and is comparable to fine-grained sense distinctions in literature. The t-test evaluates whether the improvement over baseline, if any, is significant.

combined per-instance precision is 61.1%. Given the degree of polysemy, it compares favorably with a number of SENSEVAL results where the average degree of polysemy for fine-grained sense distinctions of verbs is much lower (e.g. 6.3 in SENSEVAL-3 English Lexical Sample task and 7.8 in SENSEVAL-1), with best systems achieving 70-73% precision on fine-grained lexical sample tasks.

Baseline performance figures and improvement over baseline for each of the verbs likewise need to be considered (see Table 1). Comparing the majority value baseline and the best precision in the SENSEVAL-3 English Lexical Sample task, we would get the estimate of the average improvement over baseline for the best-performing system in SENSEVAL-3 at 32% (with majority value baseline at 55.2% and best precision of 72.9%). In the present experiments, the average improvement over baseline estimated in the same manner gives the improvement of 34%.

The information gain-based feature weeding as implemented here fails to consistently improve the performance for all the verbs; however, five out of seven verbs achieve improvement in precision when the filtered feature set is used instead of the full set of features. The full feature sets for “seek” and “back” perform slightly better than the filtered ones.

5. References

Eneko Agirre and David Martínez. 2004. The Basque Country University system: English and Basque tasks. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 44–48, Barcelona, Spain, July. Association for Computational Linguistics.

T. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text. *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Canary Islands, May 2002*, pages 1499–1504.

Gerard Escudero, Lluis Màrquez, and German Rigau. 2004. TALP system for the English lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 113–116, Barcelona, Spain, July. Association for Computational Linguistics.

Z. Harris. 1985. Distributional structure. In J. Katz, editor, *Philosophy of Linguistics*, pages 26–47. Oxford University Press, New York.

Y. K. Lee, H. T. Ng, and T. K. Chia. 2004. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 44–48, Barcelona, Spain, July. Association for Computational Linguistics.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The senseval-3 english lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain, July. Association for Computational Linguistics.

G. Miller and W. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

J. Pustejovsky and P. Hanks. 2001. Very Large Lexical Databases: A tutorial. *ACL Workshop, Toulouse, France*.

J. Pustejovsky, P. Hanks, and A. Rumshisky. 2004. Automated Induction of Sense in Context. In *COLING 2004, Geneva, Switzerland*, pages 924–931.

J. Pustejovsky, C. Havasi, R. Sauri, P. Hanks, A. Rumshisky, and J. Castano. 2006. Towards a generative lexical resource: The Brandeis Semantic Ontology. In *LREC 2006, Genoa, Italy*.

A. Rumshisky, P. Hanks, C. Havasi, and J. Pustejovsky. 2006. Towards a generative lexical resource: The Brandeis Semantic Ontology. In *FLAIRS 2006, Melbourne Beach, Florida, USA*.