Automatic Detection of Orthographic Cues for Cognate Recognition

Andrea Mulloni and **Viktor Pekar**
Research Group in Computational Linguistics
HLSS, University of Wolverhampton
MB114 Stafford Street
Wolverhampton, WV1 1SB, United Kingdom
{andrea2|v.pekar}@wlv.ac.uk

Abstract

Present-day machine translation technologies crucially depend on the size and quality of lexical resources. Much of recent research in the area has been concerned with methods to build bilingual dictionaries automatically. In this paper we propose a methodology for the automatic detection of cognates between two languages based solely on the orthography of words. From a set of known cognates, the method induces rules capturing regularities of orthographic mutations that a word undergoes when migrating from one language into the other. The rules are then applied as a preprocessing step before measuring the orthographic similarity between putative cognates. As a result, the method allows to achieve an improvement in the F-measure of 11,86% in comparison with detecting cognates based only on the edit distance between them.

1. Introduction

Present-day machine translation technologies crucially depend on the size and quality of lexical resources. MT software typically comes together with a wide range of specialized lexicons, as well as tools for their customization. In practice, however, having a lexical repository truly appropriate for a specific task is problematic: It is still very difficult to find a specific lexicon for the required language pair and/or topic area. Facing this situation, a lot of recent NLP research has been focusing its efforts on finding ways to automatically induce lexical knowledge from corpora.

The detection of cognates, i.e. words that have similar spelling and meaning in different languages, such as English *government* and French *gouvernment*, proved very helpful for bilingual lexicon compilation and related tasks. Cognates account for a considerable amount of unique words in many lexical domains, notably technical texts. The orthographic similarity of cognates can be exploited in different tasks involving recognition of translational equivalence between words, such as statistical machine translation and bilingual terminology compilation. In these tasks, the orthographic similarity of cognates can compensate for the insufficiency of other kinds of evidence about translational equivalency of words.

This paper proposes a methodology for the induction of orthographic rules for cognate recognition in a given language pair. Because the rule induction procedure is fully automatic, it allows for robust large-scale acquisition of cognates and is easily portable across different language pairs and knowledge domains.

The rest of the paper is organized as follows. In Section 2 we describe the previous work that has been carried out on cognate detection together with some of its practical applications, while in Section 3 we present our approach and deal in greater detail with our learning and testing algorithms. Once the proposed methodology has been outlined, we step through an evaluation method we devised and report on the results obtained as specified in Section 4, for then tackling tasks and future challenges in Section 5.

2. Previous Work

Generally speaking, there have been two major approaches to the problem of identifying cognates in two languages. The first one is based on the manual design of rules describing how orthography of a borrowed word should change once it has been introduced into the other language. Koehn and Knight (2002) expand a list of English-German cognate words by applying well-established transformation rules (e.g. substitution of *k* or *z* by *c* and of *–tät* by *–ty*, as in Ger. *Elektizität* – Eng. *electricity*). They also noted that the accuracy of their algorithm increased proportionally with the length of the word, since the accidental coexistence of two words with the same spelling but with different meanings (also called false friends, e.g. Eng. *art* – Ger. *Art* 'type, kind') is much higher in shorter words.

The second approach is to rely on a certain measure of the spelling similarities between the two words involved. The most eminent approach to orthographic similarity is edit distance – also known as Levenshtein distance – which corresponds to the minimum number of edit operations (substitution, deletion and insertion) required to transform one word into another (Levenshtein, 1965). Cognate recognition using edit distance has been proposed by Mann and Yarovsky (2001), who try to induce translation lexicons between cross-family languages via third languages. Lexicons are then expanded to intra-family languages by means of cognate pairs and cognate distance. Related techniques are the longest common subsequent ratio, which counts the number of letters shared by two strings and divides it by the length of the longest string (Melamed, 1995), and a method developed by Danielsson and Mühlenbock (2000), which associates two words by calculating the number of matching consonants, allowing for one mismatched character. A further interesting spin-off has been mentioned by Kondrak (2004), who stresses the importance of genetic cognates by comparing the phonetic similarity of lexemes with the semantic similarity of the glosses.

Possible concrete applications of cognate detection techniques are described by Simard *et al.* (1992) and Melamed (1999), who aim to implement cognate recognition for sentence alignment purposes in bilingual

corpora. Notably, (Melamed, 1999) proposes a more accurate cognate criterion driven by approximate string matching. The induction of translation lexicons has been analyzed in detail by many researchers, among which Mann and Yarowsky (2001), who demonstrated the power of (weighted) edit distance compared to HMM and stochastic transducers, while a practical application of cognates has been implemented by Kondrak and Dorr (2004), who report that orthographic similarity measures are of great help in the identification of drug names.

A general overview and a basic comparison of statistical techniques for cognate detection is delivered by Inkpen *et al.* (2005), who address the problem of automatic classification of word pairs as cognates or false friends and analyze the impact of applying different features through machine learning techniques.

3. Proposed Approach

While a high degree of orthographic similarity indicates the two words belonging to different languages are cognates, many unrelated words may have great similarity in spelling (e.g. Eng. *black* and Ger. *Block*). And vice versa, two words may be cognate, but their orthographies may have little in common (e.g., Eng. *cat* and Ger. *Katze*). The approach we propose is based on the assumption that between two given pairs of languages there are certain regularities in which the spelling of a word changes once it is borrowed from one language into the other. Taking into account these regularities – termed "orthographic cues" – before measuring orthographic similarity between them, can greatly facilitate recognition of cognates between two languages. The proposed method aims to maximize the number of detected cognates in a corpus by implementing an algorithm which learns orthographic cues from known cognates, for then using such cues to produce a more comprehensive cognate list. In the following we describe an algorithm that learns cognate detection rules from a list of known cognates (Section 3.1) and an algorithm for applying the induced rules to pairs of words prior to measuring orthographic similarity between them (Section 3.2).

3.1 Learning Algorithm

The learning algorithm involves three major steps: (a) the association of edit operations to the actual mutations that occur between two cognate words; (b) the extraction of candidate rules; (c) the assignment of a statistical score to the extracted rules in order to identify only the most reliable ones.

Input: C, a list of English-German cognate pairs $\{e,g\}$ χ^2_{cutoff}, a threshold on the association score between e and g
Output: R, a set of mutation rules

1 **for** c in C **do**:
2 determine edit operations to arrive from e to g
3 from each edit operation, create a candidate rule r
4 **end**
5 **for** r_c in R_c **do**
6 compute χ^2 (r_c)
7 **if** χ^2 (r_c) > χ^2_{cutoff} : output r_c as r Î R
8 **end**

Figure 1. The rule induction algorithm

3.1.1 Edit Operation Association

Figure 1 describes the specific steps of the algorithm. The algorithm takes as an input a list of cognate pairs C in two languages, each consisting of an English word e and a German word g. The output of the algorithm is a set of rules R. In the beginning, two procedures are applied to the data: (a) edit operations between the two strings of the same pair are identified (line 2 in the algorithm); (b) the normalized edit distance (NED) between each pair is calculated in order to assign a score to each cognate pair. NED is calculated by dividing edit distance (ED) by the length of the longer string. NED – and normalization in general – allows for more consistent values, since we noticed that when applying standard ED, word pairs of short length (2 to 4 words each) would be more prone to be included in the cognate list even if they are actually unrelated (e.g. at/an, hag/hexe) Sample output of this step is shown in Figure 2.

```
toilet/toilette
t     |o     |i     |l     |e     |t     |      |
t     |o     |i     |l     |e     |t     |t     |e
MATCH |MATCH |MATCH |MATCH |MATCH |MATCH |INS   |INS

tractor/traktor
t     |r     |a     |c     |t     |o     |r
t     |r     |a     |k     |t     |o     |r
MATCH |MATCH |MATCH |SUBST |MATCH |MATCH |MATCH

absolute/absolut
a     |b     |s     |o     |l     |u     |t     |e
a     |b     |s     |o     |l     |u     |t     |
MATCH |MATCH |MATCH |MATCH |MATCH |MATCH |MATCH |DEL
```

Figure 2. Edit operation association

3.1.2 Candidate Rule Identification

At the next stage of the algorithm, we extract a candidate rule c_r from each edit operation of each word pair in the training data. Each candidate rule consists of two letter n-grams. To construct it, for each edit operation detected we use k symbols on either side of the edited symbol in both e and g. The left-hand side refers to the English n-gram, while the right-hand side corresponds to the same n-gram in German with the detected mutations. Figure 3 illustrates rules detected in this manner.

Candidate rules are extracted using different values of k for each kind of edit operations, each value having been set experimentally. Substitution rules are created without considering the context around the letter being substituted, i.e. taking into account only the letter substitution itself, while deletions and insertions are sampled with k symbols on both sides. After extensive testing, k has been empirically set to two: this decision was supported by the fact that longer "rules" are less frequent than shorter "rules", but they are nonetheless more precise. In fact, because of the task at stake and the further areas we want to apply the algorithm to, we were somewhat more inclined towards obtaining higher precision rather than higher recall.

Furthermore, when sampling the candidate rules we were interested in highlighting the position of the rule itself within the string, and added an extra character (#) to mark word boundaries (e.g. #fie/#fe, sh#/sch#).

Rule	Chi-square score
c/k	386.8783163
d/t	345.6994357
ary#/är#	187.9303777
my#/mie#	187.9303777
hy#/hie#	187.9303777
gy#/gie#	172.5103846
ty#/tät#	167.5170499
et#/ett#	162.5970468
sh#/sch#	157.7503753
ive#/iv#	148.2770267

Figure 3: Top 10 rules detected by the algorithm along with the associated chi-square scores. The hash symbol stands for a word boundary.

3.1.3 Candidate Rule Scoring

At this stage, statistical scores are assigned to each unique candidate rule that has been extracted (step 6). After exploring different scoring functions (Fisher's exact test, chi-square, odds ratio and likelihood ratio), we chose to use chi-square for measuring the strength of the association between the left-hand side and the right-hand side of the candidate rule. Once every candidate rules has been associated to a chi-square value, we filter out the candidates that fall below a specific threshold on the chi-square value, thus outputting the final rules.

3.2 Testing algorithm

The learning algorithm has provided us with a set of rules which account for the orthographic behaviour of words between a source language (English) and a target language (German). The second part of the algorithm (i.e. the testing algorithm) now tries to deploy this kind of information (input) in order to create an expanded list of the cognates entailed in an English-German dictionary (final output), where the dictionary contains both cognate and non-cognate pairs.

Once the input data is made available, we proceed to apply the rules to each entry, that is to substitute relevant n-grams in the rules with their counterpart in the target language. NED is then computed for every pair, and the pairs which fall within a specific cut-off are added to the cognate list. A case in point is represented by the entry "*electric*/*elektrisch*": the original NED is 0.300, but if we apply the rules *"c/k"* and *"ic/isch"* detected earlier in the algorithm, the new NED is 0.000. Our algorithm has now produced what we originally aimed at: an "expanded" list of cognates, that is a list that includes also cognates not detected by means of a plain surface analysis.

Input: *D*, a dictionary
R, a set of mutation rules
NED_{cutoff} a NED threshold for cognates
Output: a list of English-German cognate pairs in D
1 **for** *d* in *D* **do**:
2 determine *e*¢ by applying relevant rules to *e*
3 calculate NED(*e*¢, *g*)
4 **if** NED(*e*¢, *g*) > NED_{cutoff}: output {*e*,*g*} as a cognate pair
5 **end**

Figure 4: The rule induction algorithm

4. Evaluation

The experimental task consisted in detecting pairs of English and German cognates among non-cognate equivalents. While some cognates can be reliably discovered merely by measuring the difference in spelling between them (words with identical spelling are very likely to be cognates), the challenge was to maximize the number of detected cognates by acquiring orthographic cues and using them to predict how cognates with greater differences in spelling would appear.

4.1 Data

The method was evaluated on a small (British) English-German dictionary consisting of 10,239 entries in a double column format[1]. The dictionary included general terminology and was not bound to specific knowledge domains.

For the task described in this paper, the original input data was split in two datasets: 90% was used for training and 10% for testing.

To obtain a gold standard for automatic evaluation, two linguists were asked to manually compile a list of the true cognates out of the input data. In situations where agreement between the experts was not reached, we decided to exclude the pair from the gold standard.

4.2 Task Description

The task of this exercise was to separate cognates from non-cognates in a bilingual English-German lexicon. This was achieved by identifying orthographic mutations in the source and target language on the basis of a first list of cognates induced from the original input data. Mutations are formulated into rules, then applied to the original input data in order to expand the cognate list detected by the first iteration.

As far as thresholds are concerned, we chose to evaluate our algorithm by testing a scenario where the chi-square cut-off for candidate rule validation was set to 50.

4.2.1 Cognate List Extraction

The first task to be performed was the extraction of the cognate list from the input data. This was achieved by calculating the NED of each entry in order to highlight possible candidates for cognateness. It should be mentioned that the problem of false friends will not be considered here, since it will be as addressed by a specific module at a later stage.

4.2.2 Rule Extraction

Once the basic cognate list was produced, we proceeded to extract the candidate rules and to validate them as described in Section 3.1. Depending on different threshold settings, the algorithm outputs highly varying number of rules (from 0 to 217).

4.2.3 Cognate List Expansion

At this stage, we traversed the test data and applied the newly discovered rules, for then recompute NED of the modified pairs. Figures 5 shows an example of cognate detection applying a chi-square cut-off of >= 50

[1] http://www.june29.com/IDP

for the rule selection task and different NED cut-offs for the rule identification process. Weighting of precision and recall was set equally (a = 0.5).

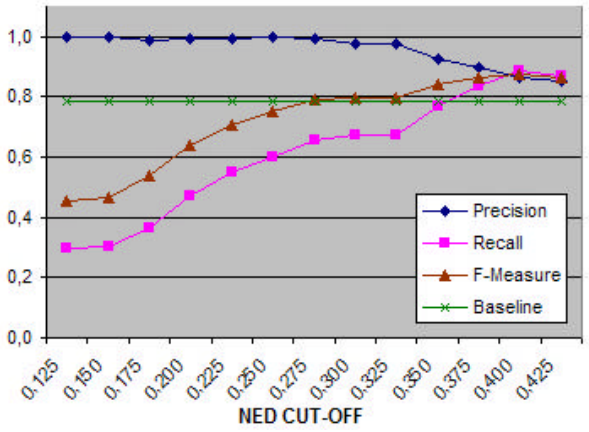

Figure 5. Scores of cognate detection using rules. The baseline value refers to the highest F-measure obtained applying ED to the same test data.

The baseline used for this experiment was set to 0.784, which corresponded to the highest F-measure obtained applying various ED cut-offs[2] to the same amount of randomly generated test data used for the testing of the cognate detection algorithm with rules.

4.2.4 Results

A quite positive effect of our approach is that precision did not seem to suffer too much when increasing the NED cut-off, while recall increased as expected. This peculiarity translates into a F-measure which outperforms the baseline starting from a NED cut-off of 0.275 and keeps improving up to the 0.400 cut-off, which is in fact our best achievement to date (F-measure = 0.877, which corresponds to an improvement of 11,86% on the baseline).

As far as detection errors are concerned, we noticed that single-letter rules (substitutions) tended sometimes to consider as cognates pairs that didn't belong to such category. A possible solution to this problem could be to introduce some kind of weighting for the rules detected, but this kind of approach requires values that we considered too arbitrary and that would anyway vary considerably across language pairs. We therefore decided to take this minor flaw on board and to try and counteract to this side-effect by expanding the number of rules that subsume the single-letter rules. The implementation of this approach is still work in progress.

5. Conclusions and Future Work

We proposed an algorithm for the detection of cognates from two different languages. Such methodology allows for the creation of a seed lexicon from unannotated text, which can then be easily integrated into many NLP applications.

Our algorithm has shown promising results, indicating that the discovered orthographic cues can considerably increase the number of cognates detected by means of measuring the similarity in their spelling.

We evaluated our algorithm on an English/German dictionary taking into account two different scenarios and producing a respectable F-measure with high levels of precision. Furthermore, considerable improvements have been reported after the implementation of "translation rules" detected from the original input data, which underlie the validity and effectiveness of our approach.

Future work will focus on the development of a method for the automatic computation of the NED threshold and the n-gram length of the rules, as well as on the evaluation of the method by using comparable corpora as input.

Finally, the extension of this approach to language pairs which have different degrees of typological relatedness is also a issue we would like to focus on in the near future.

References

Danielsson, P., Muehlenbock, K. (2000). Small but Efficient: The Misconception of High-Frequency Words in Scandinavian Translation. *In: Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, Springer Verlag, London, 158-168.

Inkpen, D., Frunza, O., Kondrak, G. (2005). Automatic Identification of Cognates and False Friends in French and English. *In: Proceedings of the International Conference Recent Advances in Natural Language Processing*, 251-257.

Koehn, P., Knight, K. (2002). Estimating Word Translation Probabilities from Unrelated Monolingual Corpora Using the EM Algorithm. *In: Proceedings of the 17th AAAI*, 711-715.

Kondrak, G (2004). Combining Evidence in Cognate Identification. *In: Proceedings of Canadian AI 2004: 17th Conference of the Canadian Society for Computational Studies of Intelligence*, 44-59.

Kondrak, G., Dorr, B. (2004). Identification of Confusable Drug Names: A New Approach And Evaluation Methodology. *In: Proceedings of COLING 2004: 20th International Conference on Computational Linguistics*, 952-958.

Levenshtein, V. (1965). Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Mann, G., Yarowsky, D. (2001). Multipath Translation Lexicon Induction via Bridge Languages. *In: Proceedings of NAACL 2001: 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, 151-158.

Melamed, D. (1999). Bitext Maps and Alignment via Pattern Recognition. *Computational Linguistics*, 25(1), 107-130.

Simard, M., Foster, G., Isabelle, P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. *In: Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, 67-81.

[2] Values from 0 to 5 have been computed: In this case the best performing ED cut-off is 3.