Word Transformation Heuristics Against Lexicons for Cognate Detection

Alexandra L. Uitdenbogerd
RMIT University
GPO Box 2476, Melbourne VIC 3001 Australia
sandrau@rmit.edu.au

Abstract

One of the most common lexical transformations between cognates in French and English is the presence or absence of a terminal "e". However, many other transformations exist, such as a vowel with a circumflex corresponding to the vowel and the letter s. Our algorithms tested the effectiveness of taking the entire English and French lexicons from Treetagger, deaccenting the French lexicon, and taking the intersection of the two. Words shorter than 6 letters were excluded from the list, and a set of lexical transformations were also used prior to intersecting, to increase the potential pool of cognates. The result was 15% above the baseline cognate list in the initial test set, but only 1% above it in the final test set. However, its accuracy was consistant at about 37% for both test sets.

1 Credits

2 Introduction

When assessing readability of English for French native speakers, or French for English native speakers, the cognates — words with similar appearance and meaning — tend to be relatively difficult words, making traditional readability measures less effective than a simple average words per sentence (Uitdenbogerd, 2005). While most words that look similar in French and English are cognates, some commonly occurring words that look similar, such as "a", "as", and "an", tend to be false friends. Other words are partial cognates, having a similar meaning only in some situations (Wang and Sitbon, 2014). Our approach ignored all context and focussed on simple heuristics. All approaches were based on taking the intersection of the French and English lexicons of the Treetagger Part of Speech tagger (Schmid, 1994), after applying a set of lexical transformations. The submission was a "quick and dirty hack" implemented on little sleep during conference attendance, and merely demonstrates that a simple heuristic-based algorithm can beat a manually curated list of cognates, albeit not by much. However, the approach should perform better than demonstrated in the ALTW challenge if applied more comprehensively.

3 Algorithms

The first "quick and dirty" baseline algorithm (**Algorithm 1**) only looks for exact matches once case and accents are preprocessed:

1. Replace the accented letters in the French lexicon with unaccented letters. For example, replace "ê" and "é" with "e".
2. Casefold and remove punctuation from the words in the source and cognate file.
3. Take the intersection of the sanitised source and cognate file words.
4. All words in the text that are in the intersection are counted as cognates.

Algorithm 2 discards any Algorithm 1 words of 5 letters or less as false friends. Common false friend English words that become discarded as a result include: "a", "as", "an". However, the following cognates are also discarded: "ah", "oh".

Algorithm 3 uses lexical transformations to the French lexicon list before intersecting with the English lexicon. It is done with and without plurals. The list is based on observation on reading a French text. While there are many transformation rules, they are not comprehensive. In particular, words in which multiple transformations are required to create an exact match are likely to be missed. Figure 1 shows the regular expression-based substitutions applied to the file.

Alexandra Uitdenbogerd. 2015. Word Transformation Heuristics Agains Lexicons for Cognate Detection . In *Proceedings of Australasian Language Technology Association Workshop*, pages 142−144.

Table 1: Training Set Statistics

	Actual	Algorithm 3
Cognates	1670	703
Non-Cognates	9425	10392
Total Words	11095	
Proportion of Cognates	0.150	0.063
Precision		0.661
Recall		0.278
F1		0.390

Table 2: ALTW Challenge 2015 Results

Team	Public	Private
LookForward	0.70478	0.77001
Little Monkey	0.67118	0.71415
MAC	0.59927	0.66857
toe_in (Alg. 4 lex.tr.-shrt)	0.37019	0.36697
Alg. 3 (lex. trans.)	0.31394	0.37272
Alg. 2 (Exact - shrtwrds)	0.23583	0.23908
Baseline	0.22951	0.34158
Alg. 1 (Exact Match)	0.22107	0.27406
Stemmed lexicon match	0.11347	0.14116

Algorithm 4 combines Algorithm 3's lexical transformations with discarding words that are 5 letters or fewer in length.

We also tried stemming but the result was half the accuracy of the original baseline. The final submission used the transformations as well as discarding words of 5 letters or less.

4 Results

Table 1 shows the precision, recall and F1 measure for the training data set.

Table 2 shows the overall results for the ALTW challenge. As can be seen, our entry (toe_in) had the most consistent performance across the two test sets. Of our submissions, Algorithm 3 performed the best on the public test dataset. The best private data submission was a version of Algorithm 3 that didn't discard short words.

In a post-analysis using the training data set we looked at the effect of varying the minimum word length for cognates, holding the base word list constant. Table 3 shows the effect on precision, recall and F measure. Precision increases as the minimum word length is increased, and recall decreases. The sweet spot in the training data set is to discard words that are 4 letters long or less.

Table 3: The effect of minimum word length on cognate detection reliability

Min Length	Precision	Recall	F measure
3	.457	.346	.393
4	.457	.346	.393
5	.579	.323	.414
6	.658	.266	.378
7	.711	.198	.309

5 Discussion

The experimental results demonstrated that a lexically transformed French lexicon intersected with an English lexicon with the shortest words discarded can be a substitute for a manually curated list of cognates, achieving 1 to 15% higher accuracy on the given test sets. A more comprehensive set of lexical transformations is likely to give a slightly higher accuracy again.

However, as the ALTW challenge results demonstrate, this context-free, heuristic approach has only about half the accuracy of the best technique.

Acknowledgments

Some text processing code was written by Aidan Martin.

Support for this project has been provided by the Australian Government Office for Learning and Teaching. The views in this project do not necessarily reflect the views of the Australian Government Office for Learning and Teaching.

References

A. L. Uitdenbogerd. 2005. Readability of French as a foreign language and its uses. In A. Turpin and R. Wilkinson, editors, *Australasian Document Computing Symposium*, volume 10, December.

H. Schmid. 1994. Probabilitic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

H. Wang and L. Sitbon. 2014. Multilingual lexical resources to detect cognates in non-aligned texts. In G. Ferraro and S. Wan, editors, *Proceedings of the Australasian Language Technology Association Workshop*, pages 14–22, Melbourne, Victoria, Australia, November. ALTA.

```
grep "e$" $1 | sed "s/e$//"
grep "ait$" $1 | sed "s/ait$/ed/"
grep "aient$" $1 | sed "s/aient$/ed/"
grep "gue$" $1 | sed "s/gue/g/"
grep "é$" $1 | sed "s/é$/y/"
grep "euse$" $1 | sed "s/euse/ous/"
grep "eux$" $1 | sed "s/eux/ous/"
grep "ique$" $1 | sed "s/ique/ic/"
grep "^dé$" $1 | sed "s/^dé/dis/"
grep "ont$" $1 | sed "s/ont$/ount/"
grep "ond$" $1 | sed "s/ond$/ound/"
grep "ant$" $1 | sed "s/ant$/ing/"
grep "ain$" $1 | sed "s/ain$/an/"
grep "aine$" $1 | sed "s/aine/an/"
grep "re$" $1 | sed "s/re$/er/"
grep "ment$" $1 | sed "s/ment$/ly/"
grep "é$" $1 | sed "s/é$/ated/"
grep "é$" $1 | sed "s/é$/ed/"
grep "ée$" $1 | sed "s/ée$/ated/"
grep "ée$" $1 | sed "s/ée$/ed/"
grep "i$" $1 | sed "s/i$/ished/"
grep "ir$" $1 | sed "s/ir$/ish/"
grep "er$" $1 | sed "s/er$/e/"
grep "ô" $1 | sed "s/ô/os/"
grep "ê" $1 | sed "s/ê/es/"
grep "î" $1 | sed "s/î/is/"
grep "ement$" $1 | sed "s/ement$/ly/"
grep "eusement$ $1| sed "s/eusement$/ously/"
grep "isme$" $1 | sed "s/isme$/ism/"
grep "if$" $1| sed "s/if$/ive/"
grep "asse$" $1 | sed "s/asse$/ace/"
grep "eur$" $1 | sed "s/eur$/or/"
grep "eur$" $1 | sed "s/eur$/er/"
grep "eur$" $1 | sed "s/eur$/our/"
grep "^é" $1 | sed "s/^é/es/"
grep "^é" $1 | sed "s/^é/s/"
grep "oût" $1 | sed "s/oût/ost/"
grep "^av" $1 | sed "s/^av/adv"
grep "^aj" $1 | sed "s/^aj/adj"
grep "elle$ $1 | sed "s/elle$/al/"
grep "ette$" $1 | sed "s/ette$/et/"
grep "onne$" $1 | sed "s/onne$/on/"
grep "quer$" $1 | sed "s/quer$/cate/"
grep "ai" $1 | sed "s/ai/ea/"
grep "^en" $1 | sed "s/^en/in/"
grep "ier$" $1 | sed "s/ier$/er/"
```

Figure 1: The set of lexical transformations applied to the French lexicon prior to intersection with the English lexicon. "$1" is the file containing the French lexicon.