LazyFriends: A knowledge-rich false friends classifier

Tim Feuerbach
uni@spell.work

Abstract

We present *LazyFriends*, a simple classifier that distinguishes between English/German cognates and false friends. Using a manually crafted bilingual dictionary, it placed first in a student shared task on false friends detection by achieving 91.88% accuracy.

1 Introduction

Words from two languages that have a similar spelling or pronounciation pose as a two-edged sword to the language learner. On the one hand, the vocabulary of the target language can be learned more quickly, since word meanings can be derived from the learner's knowledge of their own language; e.g., it is easy to memorize that the French *liberté* translates to *liberty*. On the other hand, similarity may be misleading. For example, the German *Labor* does not mean *labor* in English, but rather *laboratory*. The former pairs are called "true cognates", the latter "false friends" (Mitkov et al., 2007). Recognizing them automatically can be used to correct common errors made by language learners.

The eNLP 2015 Shared Task was a student competition in the context of the lecture "Natural Language Processing and e-Learning" at Technische Universität Darmstadt. It dealt with the classification of English/German cognates and false friends. Performance of the competing systems was evaluated on a test set undisclosed prior to publication of the final results. The systems were given a list containing 1120 pairs of English and German words and they had to decide whether they were cognates or false friends. If they were cognates in one sense but false friends in the other, the words were considered cognates in the gold standard.

In this paper, we present our participating system *LazyFriends*, which uses a simple dictionary lookup to perform the classification. In Section 2, we introduce the classification algorithm. We discuss the Shared Task results in Section 4, where we also perform an error analysis. Finally, Section 5 concludes.

2 Method

```
Given:
dict: an English-German dictionary
e, g: the English and German word to classify
$dict_{en}$: set of English words in the dictionary
$dict_{de}$: set of German words in the dictionary

CLASSIFY(E, G):
  if (e, g) in dict
    ↑ COGNATE
  else
    if e not in $dict_{en}$ and g not in $dict_{de}$
      ↑ COGNATE
    else
      ↑ FALSE FRIEND
```

Figure 1: *LazyFriends* algorithm

Our system exploits the fact that there existed only two possible classes, cognate or false friend, and no class "unrelated". This renders orthographic similarity metrics, for example explored by Inkpen et al. (2005), obsolete, since we do not need to know whether two words could be confused – this is automatically assumed to be true for all pairs. Classifying those pairs therefore boils down to finding out whether an English and a German word share a similar meaning across at least one sense. We used the algorithm depicted in Figure 1, which is able to solve the problem perfectly given an oracle bilingual lexical resource. The system is deterministic and requires no training data.

If our dictionary contains neither the German nor the English word, we fall back to the cognate decision. This has increased accuracy on our development set by 1 point in comparison to a method that falls back to the "false friends" class. We assume English loan words like *selfie* in the German language to be responsible for this. They are easily identifiable as cognates, but not yet included in dictionaries.

3 Data

There are three possible ways to obtain a bilingual dictionary:

- Let humans create the entries. This ensures high precision, but implies low coverage, slow updates, and high costs.
- Generate the entries in an unsupervised fashion from an aligned corpus. This method introduces serious noise, but benefits from high recall.
- A combination of the previous two; enrich an existing manually crafted lexicon automatically using a corpus.

We decided to use conventional dictionaries created by humans. since we aimed for high precision. We assumed the test data to contain a lot of "classic" examples of false friends, which are covered even by most dictionaries quite well. Three participants of the Shared Task used Uby (Gurevych et al., 2012), which combines multiple supervised resources like WordNet (Miller, 1995) or Wikipedia, but this involves automatic alignment of words from two languages, which reduces precision as it may lead to false friends being incorporated as valid translations.

There is only a small number of machine-readable dictionaries available for free. We originally intended to use the dict.cc dictionary[1], as we found it to cover a large number of words in our everyday use; however, their license does not permit the distribution of the data alongside the program. Instead, we used two other dictionaries: Ding[2] (maintained by Frank Richter), which was once the base for dict.cc, and FreeDict[3] (Horst Eyermann, Michael Bunk et al.).

FreeDict provides the data in XML TEI format, which makes it easy to use in an application, while Ding entries are formulated as they would be in a dictionary for humans, with minimal indications of separation:

```
Aasfliege {f} [zool.] | Aasfliegen {pl} ::
carrion fly; fleshfly; flesh fly | carrion
flies; fleshflies; flesh flies
```

We therefore had to first parse the data using regular expressions. For both dictionaries, we left multiwords in, although they were not part of the task. We used both the English-German and German-English version of FreeDict, since there was a small number of non-overlaps. We obtained 124 070 entries from FreeDict and 603 840 entries from Ding.

For evaluation during development, we used lists of English/German cognates and false friends from about.com[4], which we made machine readable with a combination of regular expressions and manual corrections. From this, we randomly drew a development set with an equal number of false friends and cognates.

4 Results and error analysis

We experimented with different combinations of dictionaries and fallback settings, and found *LazyFriends* to perform best when using only the Ding dictionary and falling back to cognates in case of unknown words. This is the setup we submitted.

The results of our system on the test data in terms of accuracy, as well as results from competing sys-

[1] http://www.dict.cc/

[2] https://www-user.tu-chemnitz.de/~fri/ding/

[3] http://freedict.org/de/

[4] http://german.about.com/library/blcognates_A.htm, http://german.about.com/library/blfalsef.htm

System	Accuracy	FF/C Mc. Ratio
Baseline	80.36	–
LazyFriends-Ding	**91.88**	0.32
System 2	87.23	0.88
System 3	38.84	0.10
System 4	86.88	0.32
System 5	67.41	0.04
System 6	84.64	0.19
System 7	60.09	0.51
LazyFriends-Ding, FF fallback	90.80	0.27
LazyFriends-FreeDict	83.30	0.21
LazyFriends-Ding&FreeDict	**92.14**	0.80

Table 1: Accuracy and false friend/cognate misclassification ratio of the participating systems compared to a baseline classifying all pairs as false friends.

tems, are displayed in Table 1.[5] Also included is a baseline that classifies all pairs as false friends (the majority class). Furthermore, we report the ratio of false friends to cognate misclassifications, calculated as $\frac{\text{\# misclassified false friends}}{\text{\# misclassified cognates}}$. For example, a number of 1.0 means that the number of mistakes made by the system is equal for both classes.

As can be seen, our system beat both the baseline and the task competitors. We could have even achieved a higher accuracy if we had submitted the approach with both dictionaries. However, our development evaluation showed high noise arising from the use of FreeDictionary, with more false friends classified as cognates. Except for System 2 and *LazyFriends* with both dictionaries, all systems misclassified about three times more cognates as false friends than vice versa. In case of approaches like ours based on lexical resources, this indicates that the majority of errors was caused by missing translation entries.

We manually inspected the 91 errors our system made on the test set. 23 errors were incorrect gold labels due to multiple senses, from which one sense pair is a false friend. For instance, mistaking the English *billion* for the German *Billion* (correct translation would be *Milliarde*) is a common mistake even among journalists, but this pair is actually a cognate in dated British English. [6] Another example is *circle – Zirkel*, which should be *divider* when referring to the drawing aid, but is a valid translation in the context of e.g. *the circle of magic*. Since the task guidelines stated that cognates overrule false friends, our system worked correctly in these cases, which were all but one of the alleged missclassifcations as cognates. 67 cognates were incorrectly classified as false friends due to missing or wrongly parsed data. Surprisingly, there was a large number of simple words one would assume to be contained in every decent dictionary, e.g. *carrot – Karrotte* or *ape – Affe*, which were not included in Ding.

5 Conclusion and outlook

We presented a simple false friend classifier that uses dictionaries crafted by humans. This approach has severe shortcomings. For once, the task was simplified due to the missing "unrelated" class. To give proper feedback, one would have to identify pairs that the learner might have confused due to similar spelling or pronounciation. Second, as could be seen from the large number of missing pairs, openly available dictionaries are far from being complete. Unsupervised approaches could increase the number of translations, but as the task results suggest, also give rise to lower precision.

[5] The architectures of the competing systems have not been disclosed by the organizers.

[6] `http://www.oxforddictionaries.com/de/definition/englisch/billion`

References

[Gurevych et al.2012] Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth. 2012. UBY - A Large-Scale Unified Lexical-Semantic Resource Based on LMF. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 580–590, April.

[Inkpen et al.2005] Diana Inkpen, Oana Frunza, and Grzegorz Kondrak. 2005. Automatic identification of cognates and false friends in French and English. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257.

[Miller1995] George A. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

[Mitkov et al.2007] Ruslan Mitkov, Viktor Pekar, Dimitar Blagoev, and Andrea Mulloni. 2007. Methods for extracting and classifying pairs of cognates and false friends. *Machine translation*, 21(1):29–53.