SBFC: An Efficient Feature Frequency-Based Approach to Tackle Cross-Lingual Word Sense Disambiguation

Dieter Mourisse[1], Els Lefever[1,2], Nele Verbiest[1], Yvan Saeys[3,4], Martine De Cock[1], and Chris Cornelis[1,5]

[1] Department of Applied Mathematics and Computer Science, Ghent University, Gent, Belgium
[2] LT3, University College Ghent, Gent, Belgium
[3] Department of Plant Systems Biology, VIB, Gent, Belgium
[4] Department of Molecular Genetics, Ghent University, Gent, Belgium
[5] Granada University, Granada, Spain
{Dieter.Mourisse,Nele.Verbiest,
Yvan.Saeys,Martine.DeCock}@UGent.be, Els.Lefever@HoGent.be,
chriscornelis@ugr.es

Abstract. The Cross-Lingual Word Sense Disambiguation (CLWSD) problem is a challenging Natural Language Processing (NLP) task that consists of selecting the correct translation of an ambiguous word in a given context. Different approaches have been proposed to tackle this problem, but they are often complex and need tuning and parameter optimization.

In this paper, we propose a new classifier, Selected Binary Feature Combination (SBFC), for the CLWSD problem. The underlying hypothesis of SBFC is that a translation is a good classification label for new instances if the features that occur frequently in the new instance also occur frequently in the training feature vectors associated with the same translation label.

The advantage of SBFC over existing approaches is that it is intuitive and therefore easy to implement. The algorithm is fast, which allows processing of large text mining data sets. Moreover, no tuning is needed and experimental results show that SBFC outperforms state-of-the-art models for the CLWSD problem w.r.t. accuracy.

1 Introduction

Word Sense Disambiguation (WSD) is the Natural Language Processing (NLP) task that consists of assigning the correct sense of an ambiguous word in a given context. Traditionally, the sense label is chosen from a predefined monolingual sense inventory such as WordNet [1]. The computational WSD task can be defined as a classification task where the possible word senses are the classes and each new occurrence of an ambiguous word is assigned to the correct sense class based on the surrounding context information of the ambiguous word.

The information that is traditionally used for WSD consists of a selection of very local context features (preceding and following words and grammatical information) and a bag-of-words feature set that reflects the presence or absence of a large set of possible content words in the wider context of the ambiguous word. As only a few of

P. Sojka et al. (Eds.): TSD 2012, LNCS 7499, pp. 248–255, 2012.

these bag-of-words features are actually present for a given occurrence of an ambiguous word, this results in very large and sparse feature vectors.

A wide range of supervised and unsupervised approaches have been proposed to tackle the WSD problem. For a detailed overview of these approaches we refer to [2]. Amongst these approaches we find all major machine learning techniques that are deployed for NLP tasks, such as memory-based learning algorithms, probabilistic models, linear classifiers, kernel-based approaches, etc. The main disadvantages of the existing classification methods are their complexity and need for tuning and parameter optimization during the training phase. As we typically work with these very large and sparse feature vectors for WSD, this leads to very complex training and classification cycles.

Cross-Lingual Word Sense Disambiguation (CLWSD) is the multilingual variant of WSD that consists of selecting the correct translation (instead of a monolingual sense label as is the case for WSD) of an ambiguous word in a given context.

This paper describes a classification algorithm that is specifically designed for the CLWSD problem, named Selected Binary Feature Combination (SBFC). We consider the CLWSD problem as a classification problem; in order to predict a correct translation of an ambiguous noun in one target language, English local context features and translation features from four other languages are incorporated in the feature vector. The main idea behind the SBFC method is that the features that occur frequently in the new instance should also occur frequently in the training instances with the predicted translation label for the new instance. The SBFC algorithm is easy to understand and fast, and can hence be used to process large-scale text mining data sets. Its advantage over other CLWSD algorithms is that it does not need tuning and is parameter independent.

The structure of this paper is as follows. Section 2 describes the data set and extracted feature set we used for our Cross-Lingual WSD experiments. Section 3 introduces our novel classification algorithm, while Section 4 provides a detailed overview of the experimental setup and results. Finally, Section 5 concludes this paper and gives some directions for future work.

2 Data

To construct the training feature set, we used the six-lingual sentence-aligned Europarl corpus that was also used in the SemEval-2010 "Cross-Lingual Word Sense Disambiguation" (CLWSD) task [3]. This task is a lexical sample task for English ambiguous nouns that consists in assigning a correct translation in the five supported target languages (viz. French, Italian, Spanish, German and Dutch) for an ambiguous focus word in a given context. In order to detect the relevant translations for each of the ambiguous focus words, we ran automatic word alignment [4] and considered the word alignment output for the ambiguous focus word to be the label for the training instances for the corresponding classifier (e.g. the Dutch translation is the label that is used to train the Dutch classifier).

For our feature vector creation, we combined a set of English local context features and a set of binary bag-of-words features that were extracted from the aligned translations. All English sentences were preprocessed by means of a memory-based

shallow parser (MBSP) [5] that performs tokenization, Part-of-Speech (PoS) tagging and text chunking. The preprocessed sentences were used as input to build a set of commonly used WSD features related to the English input sentence:

- features related to the **focus word itself** being the word form of the focus word, the lemma, Part-of-Speech and chunk information
- **local context features** related to a window of three words preceding and following the focus word containing for each of these words their full form, lemma, Part-of-Speech and chunk information

In addition to these monolingual features, we extracted a set of binary bag-of-words features from the aligned translations that are not the target language of the classifier (e.g. for the Dutch classifier, we extract bag-of-words features from the Italian, Spanish, French and German aligned translations). Per ambiguous focus word, a list of content words (verbs, nouns, adjectives and adverbs) was extracted that occurred in the aligned translations of the English sentences containing the focus word. One binary feature per selected content word was then created per ambiguous word: '0' in case the word does not occur in the aligned translation of this instance, and '1' in case the word does occur. For the creation of the feature vectors for the test instances, we follow a similar strategy as the one we used for the creation of the training instances. For the construction of the bag-of-words features however, we need to adopt a different approach as we only have the English test instances at our disposal, and no aligned translations for these English sentences. Therefore we decided to deploy the Google Translate API[1] to automatically generate a translation for all English test instances in the five supported languages.

3 Method

As described in Section 2, the CLWSD feature vectors consist of two parts. The first part, that covers the local context features, contains non-binary but discrete data. Before applying the SBFC algorithm, we used a straightforward procedure to make this data binary. For each value v of a non-binary feature f in our training set, a new binary feature is generated that is 1 if the instance has value v for f and 0 otherwise.

As a result, we have training data that consists of n ambiguous words $w_1, \ldots, w_n$, described by m binary features $f_1, \ldots, f_m$. The translation of a word w is denoted by $T(w)$, the value of a word w for a feature f_i is denoted by $f_i(w)$ and is a value in $\{0, 1\}$. We say that a feature f_i *occurs* in a word w if $f_i(w) = 1$. The task is now to predict a translation $T(t)$ for a test word t described by the binary vector $(t_1, \ldots, t_m)$.

Before the actual method is carried out, we apply a preprocessing step in order to remove the training instances with a unique translation because this very often occurs in the presence of noise. An example is given in Table 1, that lists part of the Italian training data for the ambiguous word *mood*. The word w_4 is removed, as its class label *umore* only occurs once in the training data.

The SBFC method that we designed for the CLWSD problem reflects the following ideas: a translation is a good classification label for new instances if the features that occur in the new instance occur.

[1] `http://code.google.com/apis/language/`

Table 1. Training data for translating the ambiguous word *mood* before (left-hand-side) and after (right-hand-side) preprocessing

	f_1	f_2	f_3	f_4	T
w_1	1	0	1	0	clima
w_2	0	0	1	1	clima
w_3	1	1	0	1	atmosfera
w_4	1	0	1	0	umore
w_5	1	0	0	1	atmosfera
w_6	0	0	1	0	atmosfera

	f_1	f_2	f_3	f_4	T
w_1	1	0	1	0	clima
w_2	0	0	1	1	clima
w_3	1	1	0	1	atmosfera
w_5	1	0	0	1	atmosfera
w_6	0	0	1	0	atmosfera

- I1: **at least once** in the training feature vectors associated with the same translation label
- I2: **frequently** in the training feature vectors associated with the same translation label

Note that I1 is contained in I2: if a feature occurs frequently in a feature vector, it will of course appear at least once in the feature vector. The reason why we handle these cases separately is that we want to penalize classification labels for which features that occur in the test vector never occur in training instances with this classification label. Moreover, we show in the experimental section that combining both ideas results in the best accuracies.

If we apply these hypotheses on the example in Table 1, we come to the following predictions for a given test vector (1, 0, 0, 1). *Clima* might be a good translation for this test vector, because feature f_1 and f_4 occur at least once in a training feature vector with translation *clima*. On the other hand, these features do not occur so often. By consequence, *atmosfera* is a better translation candidate, because f_1 and f_4 occur at least once in a word with translation *atmosfera*, and these features occur each twice in a word with translation *atmosfera*.

To formalize this idea, we first translate the training data into the so-called *model matrix* M. This matrix has dimensions $m \times c$ with m being the number of features and c the number of different translations appearing in the data set. The entry M_{ij} is the number of times that the ith feature occurs in a word with the jth translation in the training data. The model matrix of the example is given on the left-hand-side of Table 2.

The columns of translations that occur often will generally contain higher values and will therefore be favored in the final algorithm. To prevent this, we scale the matrix by dividing the values in the column of a translation by the times this translation occurs. This is shown in the right-hand-side of Table 2.

The scaled model matrix can now be used to predict the translation label of new test instances with vector $(t_1, \ldots, t_m)$. We first assign a score to each class label (translation) in the training data as follows: suppose the translation considered is C, then we look up the column in the scaled model matrix corresponding to this translation and count how many features occurring in the test vector have a value different from zero in this column. This measure reflects the idea in I1. To express the idea in I2, we sum the values in the column corresponding to features that occur in the test vector. The resulting score of translation C is then the product of these two measures. Finally, the translation label

Table 2. Model matrix of the example in Table 1 before (left-hand-side) and after (right-hand-side) scaling

	clima	atmosfera
f_1	1	2
f_2	0	1
f_3	2	1
f_4	1	2

	clima	atmosfera
f_1	0.5	0.67
f_2	0	0.33
f_3	1	0.33
f_4	0.5	0.67

that will be predicted by the algorithm for the given test vector will be the translation with the highest overall score.

Formally, the first measure for the jth class is given by:

$$\text{score}_1(j) = \sum_{i=1}^{m} h(M_{ij}) \cdot t_i,$$

where $h(M_{ij})$ is one if M_{ij} is different from zero and zero otherwise, and by

$$\text{score}_2(j) = \sum_{i=1}^{m} M_{ij} \cdot t_i$$

for the second measure. The final score of the jth class is then:

$$\text{score}(j) = \text{score}_1(j) \cdot \text{score}_2(j).$$

This algorithm works fast and only needs limited storage: constructing the model matrix M needs $\mathcal{O}(nm)$ operations, and the model matrix itself has dimensions $m \times c$. The original training data does not need to be stored anymore during the test phase. To classify a new test vector, $\mathcal{O}(m)$ operations are required.

Suppose the test vector in the running example is (0, 1, 1, 0). To calculate the score of the translation *clima*, we look at the first column in the scaled model matrix. Features f_2 and f_3 occur in the test vector, but only one of them has a non-zero value in the model matrix, so the first measure for *clima* is $\text{score}_1(\text{clima}) = 1$. Next, we sum the values for f_2 and f_3, which results in $\text{score}_2(\text{clima}) = 1$. The final score for the translation *clima* is $\text{score}(\text{clima}) = 1 \cdot 1 = 1$. To determine the score of *atmosfera*, we look at the second column. For both f_2 and f_3, the values are different from zero, so the first measure is $\text{score}_1(\text{atmosfera}) = 2$. Next, we sum the values for f_2 and f_3, resulting in $\text{score}_2(\text{atmosfera}) = 0.66$. The final value for translation is $\text{score}(\text{clima}) = 2 \cdot 0.66 = 1.32$, which is higher than the score for *clima*. By consequence, the translation *atmosfera* is returned by the algorithm for this given test instance.

4 Experimental Evaluation

To evaluate our classification algorithm for the five target languages, we used the sense inventory and test set of the SemEval "Cross-Lingual Word Sense Disambiguation" task. A more detailed description of the construction of the data set can be found in [7].

4.1 Experimental Set-Up

We consider three versions of the SBFC method: $SBFC_1$ (resp. $SBFC_2$) only uses $score_1$ (resp. $score_2$) to measure the quality of the class labels, while *SBFC* uses the product of the two scores. We make this distinction to show that both ideas I1 and I2 as described in Section 3 have to be taken into account. We apply the SBFC method to the CLWSD data sets for the ambiguous words *coach, education, execution, figure, letter, match, mission, mood, paper, post, pot, range, rest, ring, scene, side, soil, strain* and *test* and compare it to a baseline, three state-of-the-art CLWSD systems and Naive Bayes:

- As a **baseline**, we select the most frequent lemmatized translation that resulted from the automated word alignment.
- The **ParaSense** system [8] uses the same set of local context and translation features as described in Section 2 and a memory-based learning algorithm implemented in TIMBL [5].
- The **UvT-WSD** system [9] uses a k-nearest neighbor classifier and a variety of local and global context features and obtained the best scores for Spanish and Dutch in the SemEval CLWSD competition.
- The **T3-COLEUR** system [10] participated for all five languages and outperformed the other systems in the SemEval competition for French, Italian and German. This system adopts a different approach: during the training phase a monolingual WSD system processes the English input sentence and a word alignment module is used to extract the aligned translation. The English senses together with their aligned translations (and probability scores) are then stored in a word sense translation table, in which look-ups are performed during the testing phase.
- The **Naive Bayes (NB)** [11] classifier is a probabilistic classifier that assumes that the features are independent. We compare with this classifier because it has similarities with our new approach, that is, it is also based on the frequencies of the features, but it does not take into account the sparse nature of the data.

As evaluation metric, we use a straightforward accuracy measure that divides the number of correct answers by the total amount of test instances.

4.2 Results

Table 3 lists the average results over the different test words per language.

Table 3. Accuracy values averaged over all nineteen test words

	SBFC	$SBFC_1$	$SBFC_2$	baseline	ParaSense	UvT-WSD	T3-COLEUR	NB
Dutch	**0.70**	0.64	0.17	0.61	0.68	0.64	0.42	0.07
French	**0.75**	0.71	0.17	0.65	**0.75**	-	0.67	0.11
Italian	**0.66**	0.61	0.20	0.54	0.63	-	0.56	0.07
Spanish	**0.73**	0.67	0.23	0.59	0.68	0.70	0.58	0.07
German	**0.69**	0.67	0.22	0.54	0.67	-	0.57	0.11

As can be seen in Table 3, $SBFC_2$ does not score well. $SBFC_1$ scores better, but is outperformed by the ParaSens and UvT-WSD system. However, the SBFC method, that combines the scores used in $SBFC_1$ and $SBFC_2$, outperforms all other methods for all considered languages, and although there are some similarities with the Naive Bayes classifier, SBFC does a far better job in selecting the correct translation for a word. It is furthermore the case that, since we only use features that occur in a particular test word (i.e. features whose value is one), this algorithm works very fast on sparse binary data.

5 Conclusion

We presented the new classifier SBFC (Selected Binary Feature Combination) to tackle the Cross-lingual Word Sense Disambiguation task. The algorithm merely relies on feature frequencies and is therefore very efficient. In addition, the method is very intuitive, and hence easy to implement and comprehend; it is by consequence easy to adapt to make it more suitable for different classification data sets. Experimental results show that the SBFC algorithm outperforms state-of-the-art CLWSD systems for all five considered languages.

In future work, we will apply the SBFC method to other classification data sets and investigate alternative methods to combine the two different scores (i.e. making one score more important than the other). In addition, we will also examine other techniques to scale the model matrix.

References

1. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
2. Agirre, E., Edmonds, P.: Word Sense Disambiguation. In: Algorithms and Applications. Text, Speech and Language Technology series. Springer (2006)
3. Lefever, E., Hoste, V.: SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden, pp. 15–20 (2010)
4. Och, F., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
5. Daelemans, W., van den Bosch, A.: Memory-based Language Processing. Cambridge University Press (2005)
6. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)
7. Lefever, E., Hoste, V.: Construction of a Benchmark Data Set for Cross-Lingual Word Sense Disambiguation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA), Valletta, Malta (2010)
8. Lefever, E., Hoste, V., De Cock, M.: ParaSense or How to Use Parallel Corpora for Word Sense Disambiguation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 317–322. Association for Computational Linguistics, Portland (2011)

9. van Gompel, M.: UvT-WSD1: A Cross-Lingual Word Sense Disambiguation System. In: Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp. 238–224. Association for Computational Linguistics, Uppsala (2010)
10. Guo, W., Diab, M.: COLEPL and COLSLM: An Unsupervised WSD Approach to Multilingual Lexical Substitution, Tasks 2 and 3 SemEval 2010. In: Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pp. 129–133. Association for Computational Linguistics, Uppsala (2010)
11. Maron, M.E.: Automatic Indexing: An Experimental Inquiry. Journal of the ACM (JACM) 8(3), 404–417 (1961)