Cross Lingual Lexical Substitution Using Word Representation in Vector Space

Mohamed A. Zahran
Computer Engineering Department, Cairo University, Egypt
moh.a.zahran@gmail.com

Hazem Raafat
Computer Science Department, Kuwait University, Kuwait
hazem@cs.ku.edu.kw

Mohsen Rashwan
Electronics and Communications Department, Cairo University, Egypt
mrashwan@rdi-eg.com

Abstract

Polysemous words acquire different senses and meanings from their contexts. Representing words in vector space as a function of their contexts captures some semantic and syntactic features for words and introduces new useful relations between them. In this paper, we exploit different vectorized representations for words to solve the problem of Cross Lingual Lexical Substitution. We compare our techniques with different systems using two measures: "best" and "out-of-ten" (oot), and show that our techniques outperform the state of the art in the "oot" measure while keeping a reasonable performance in the "best" measure.

Introduction

Word sense disambiguation (WSD) is a famous problem in Natural Language Processing (NLP). It solves the problem of identifying a particular sense for polysemous word given its context. Cross Lingual Lexical Substitution (CLLS) can be regarded as a multilingual word sense disambiguation. It substitutes a word given its context in a source language with a suitable word in a target language, thus this task is inherited from Machine Translation (MT). However, many machine translation systems fail to do this task correctly mainly due to the insufficient parallel data that covers different word senses with their contexts.

We can see how the context surrounding a word is a key player in all these tasks: WSD, CLLS and MT. This suggests that the real sense for a word is formulated somehow from its surrounding context, which means that if we can represent a word as a vector in a multidimensional space as a function of its context, then words appearing in similar contexts will be related.

In this paper, we examine how to employ such vectorized representations for words to solve CLLS.

Cross Lingual Lexical Substitution (CLLS)

The problem of CLLS is described in SemEval-2010, task2 (Mihalcea et al. 2010) where given an English word in a context, it is required to find an alternative Spanish word or phrase to substitute this English word in its context. English words (headwords) were collected such that each word has instances; each instance expresses a certain sense for the word using a context. The instances are not necessarily distinct, which means that they can share translations. Four native Spanish speaker annotators were assigned to manually do the cross lingual lexical substitution for the collected dataset. Each annotator examines each headword and for each instance, he supplies as many translations as possible. Afterwards, for each instance, all Spanish words supplied by the annotators were pooled together keeping track of the frequency for each translation, so that the most frequent translation given by the annotators for any instance is most likely the correct one. The dataset is divided into test set and development set. The test set has 100 English words each has 10 instances. The development dataset has 30 English words with 10 instances each.

Related Work

Here we give brief description for the competing systems (Mihalcea et al. 2010). Two baseline systems were introduced, the first is dictionary based (DICT) and the second is both dictionary and corpus based (DICTCORP). The dictionary used is an online Spanish-English dictionary and the corpus is the Spanish Wikipedia. DICT retrieves all the Spanish translations for the English headwords and uses the first translation provided by the online dictionary as the Spanish word for all the English instances. DICTCORP sorts the retrieved Spanish translations by their frequency of occurrence in the Spanish Wikipedia corpus. UvT systems (Gompel, 2010) builds a word expert for the target words using k-nearest neighbor, then the correct translation is selected using GIZA word alignments from the Europarl parallel corpus. WLVusp (Aziz et al. 2010) uses the open machine translation framework Moses to obtain the N-best translation list for the instances, then uses English-Spanish dictionary as a filter to pick the correct translation. UBA-T and UBA-W (Basile et al. 2010) work in two steps, the first is candidate collection by retrieving and ranking candidate translations from Google dictionary, SpanishDict.com and Babylon, then UBA-T uses Google to translate instances to Spanish, while UBA-W uses parallel corpus automatically constructed from DBpedia. The second step is candidate selection, it is performed by several heuristics that use tokenization, part of speech tags, lemmatization, Spanish WordNet and Spanish translations. SWAT-E and SWAT-S (Wicentowski et al. 2010) use a lexical substitution framework. The SWAT-E system first performs lexical substitution in English, and then translates the substitutions into Spanish. SWAT-S translates the source sentences into Spanish, identifies the Spanish word corresponding to the target word, and then performs lexical substitution in Spanish.

Word Representation in Vector Space

There have been several attempts to represent individual words of a certain language in vector space so that these representations capture semantic and syntactic properties of the language. These representations can serve as a fundamental building unit to many NLP applications. Word representation is a mathematical model representing a word in space, mostly a vector. Each component (dimension) is a feature to this word, which can have semantic or syntactic meaning. We compare three new techniques to build the vector space. (Mikolov et al. 2013) proposes two new techniques for building word representation in vector space based on a neural network setting that predicts a pivot word using its context. The first technique is continuous bag of word (CBOW); this model predicts a word using a window of context. Contextual words are the inputs to the neural network and the objective of the network is to predict the pivot word. On the other hand, the second technique is the Skip-gram model. It uses the pivot word as an input to the neural network then tries to predict its contextual words within a window by maximizing the probability of the context words given the pivot word. Increasing the context window increases the model accuracy reflected in the quality of the resulting word vectors, but it increases the computation complexity. The third technique is called "GloVe" for Global Vectors (Pennington et al. 2014), while CBOW and skip-gram models can be classified as shallow window based approaches, because they represent a word in vector space as a function of its local context controlled by a window, GloVe on the other hand utilizes the global statistics of word-to-word co-occurrences in a corpus to be captured by the model. The co-occurrence matrix is used to calculate the probability of word$_i$ to appear in the context of word$_j$ $P(i \mid j)$, this probability is postulated to capture the relatedness between these words, e.g. the word "solid" is more related to "ice" than to "steam", this can be confirmed by the ratio between $P(\text{"}solid\text{"} \mid \text{"}ice\text{"})$ and $P(\text{"}solid\text{"} \mid \text{"}steam\text{"})$ to be high. Glove uses this ratio to encode the relationship between words and tries to find vectorized representation for words that satisfies this ratio, thus the model is built with the objective of learning vector representation for words capturing linear linguistic relationship between them.

System Description

The main idea behind our approach is to make use of the useful word-to-word relations in vector space to disambiguate between different senses using the context. For example, the word "bank" has the two sense; the first is the financial institution (labeled as a sense by "money") and the second is "riverside" (Table 1). Given a context for each sense, it is required to map the context to the correct sense.

Sense	Context
money	Context$_1$: He cashed a check at the bank to pay his loan.
riverside	Context$_2$: He sat by the bank of the river to watch the fish in water currents.

Table 1: Two Senses for the word "Bank" with contexts.

By examining the contexts, we can notice that the words "check", "cashed" and "loan" are strongly related to "money" more than to "riverside". Also, the words "river", "water" and "fish" show stronger relationship with "riverside" than with "finance". This relationship can be measured by using a similarity function that maps a pair of word vectors to a real number: $F(v_1, v_2) \to \mathbb{R}$ .This mapping function (similarity measure function) can be Cosine similarity, Euclidean distance, Manhattan distance, or any possible similarity measure techniques. Given a list of sense representative words e.g. "money" & "riverside" and a list of contexts, it is required to map the sense with its correct context. We propose two scoring functions that assign a score to a sense/context pair. Define $H(sense, context) \to \mathbb{R}$ the first scoring function is:

$$H_1(s_i, c_j) = \sum_{\forall\, w \in c_j} F(v(s_i), v(w)) \tag{1}$$

Where the operator $v(w)$ takes a word w and returns its vector representation, sense$_i$ is a sense representative word denoted as s_i and context$_j$ is denoted as c_j. This function calculates the similarity score for a sense and a context by accumulating the pairwise similarity score for sense$_i$ and each word in context$_j$.

Contextual Words	"money"	"riverside"
cashed	**0.293907**	0.01455
check	**0.163442**	0.039154
loan	**0.229422**	0.11067
river	0.161156	**0.595411**
water	0.245526	**0.287399**
currents	0.01496	**0.094783**
fish	0.112737	**0.271241**

Table 2: Shows the pairwise Cosine similarity between contextual words and two senses "money" and "riverside".

Table 2 shows pairwise cosine similarity scores for the senses and contexts in Table 1 using Mikolov's skip-gram word vector representations for English[1], using these scores we can apply H_1 to calculate the scores for both senses with both contexts and map the sense to the context with the highest score, thus assigning context$_1$ to "money" and context$_2$ to "riverside".

H_1('money',context$_1$)	**0.686771**
H_1('riverside',context$_1$)	0.164374
H_1('money',context$_2$)	0.534379
H_1('riverside',context$_2$)	**1.248834**

Another idea is to treat word vectors as semantic layers such that a context can be regarded as a concept formed by its individual words; each word contributes to the formation of this concept by a certain increment (semantic layer). Combining those layers should give an abstractive approximation for the concept. A simple combination for semantic layers is adding the vector representations of the words together.

$$H_2\left(s_i, c_j\right) = F(v\left(s_i\right), \sum_{\forall\, w \in c_j} v\left(w\right)) \qquad (2)$$

Applying the semantic layers idea to both contexts and calculating the similarity between the semantic layers approximated concepts with the senses will assign context$_1$ to "money" and context$_2$ to "riverside".

H_2('money',context$_1$)	**0.334804**
H_2('riverside',context$_1$)	0.081559
H_2('money',context$_2$)	0.168080
H_2('riverside',context$_2$)	**0.413259**

The rest of this section will discuss how to apply the ideas presented in the previous example to the CLLS problem. We divide our technique into three steps. First, data collection and preparations. Second, building the vector space model and finally evaluate the models and compare them to other technique.

Building word vector representation for the target language (Spanish).

Using the three models discussed (CBOW, SKIP-GRAM, and GloVe) we build word representations in vector space for Spanish. To train the models, we collect raw Spanish text from these sources made available by the open parallel corpus[2] and Wikipedia:

- MultiUN (Eisele and Chen, 2010); it is a collection of translated documents from the United Nations.
- OpenSubtitles[3] 2011, 2012, and 2013; they are a collection of movies subtitles.
- EUbookshop (Skadins et al. 2014); it is a corpus of documents from the EU bookshop.
- Europarl3 (Tiedemann, 2009); it is a parallel corpus extracted from the European Parliament web site.
- Europarl (Tiedemann, 2012); it is an improved parallel corpus extracted from the European Parliament web site.
- EMEA (Tiedemann, 2009); it is a parallel corpus made out of PDF documents from the European Medicines Agency.
- ECB (Tiedemann, 2009): it is a documentation from the European Central Bank.
- Tatoeba (Tiedemann, 2009); it is a collection of translated sentences from Tatoeba.
- OpenOffice (Tiedemann, 2009); it is a collection of documents from openoffice.org
- PHP (Tiedemann, 2009); it is a parallel corpus originally extracted from http://se.php.net/download-docs.php
- EUconst (Tiedemann, 2009); it is a parallel corpus collected from the European Constitution.
- Spanish Wikipedia dump.

We compile all these sources together and clean them from noisy characters and tags. The vocabulary size of the compiled corpus is 1.18 million words and the number of words is 2.7 billion. Next, we train the models[4] changing the window of context to 5 and 10. We refer to this window parameter later on as the model window (MWINDOW).

[1] https://code.google.com/p/word2vec/

[2] http://opus.lingfil.uu.se/

[3] http://www.opensubtitles.org/

[4] Models are available at: https://sites.google.com/site/mohazahran/data

Spanish Translation

Google Translate is used to retrieve all possible Spanish translations for each headword sorted by frequency, and to translate all instances (contexts) to Spanish. Now the CLLS problem is transformed into a mapping problem; to map between possible headword translations (acting as sense representation words) and the instances translations (acting as contexts).

Mapping Algorithm

The Spanish translations for instances are cleaned from stop-words and noisy characters and then we introduce few parameters to control the mapping algorithm:

- Similarity measure between two vectors (SIM): Cosine similarity, Euclidean distance and Manhattan distance.
- Vector normalization (NORM): The choice to normalize the word vector or not before performing a similarity measure.
- The number of output choices per instance (MAXO): The CLLS task allows for systems to output more than one suggestion for each instance, this parameter to supply a specific number of translation choices.
- The number of headword translations to consider (MAXTRNS): Each headword has more than one Spanish translation sorted by frequency. This parameter limits the number of translations to consider in order to ignore infrequent translations.
- Minimum score threshold (MINSIM): This parameter refuses to assign a context to a headword translation if their similarity score below this threshold.
- The window size around the headword (HWINDOW): To limit the words to consider in a context, we use a window around the headword translation so that words in range $[p-w:p+w]$ will only be considered as contextual words, where p is the position of headword translation and w is the window size. The intuition behind this parameter is to adjust the problem into a similar setting that was used to train our models.
- Sematic layers (SEMLAYER): to use semantic layers or not (Choose between H_1 and H_2).
- Vector averaging (AVG): In case of SEMLAYER is used, we may combine vectors by taking their average representations instead of mere addition.

Using a certain configuration of these parameters, we can transform the CLLS problem into a mapping task between sense representative words (headword word translations) and the contexts translations. It is worth noting that we removed the headword translation from all translated instances because machine translation fails to pick a correct headword translation matching the context, which means that keeping this possibly erroneous headword translation can confuse our matching algorithm.

Scoring

Two scoring metrics are used to score the systems competing in this task; “best” and “out-of-ten (oot)” (Mihalcea et al. 2010). Since systems are allowed to supply more than one translation per instance it is required to give credit to the correct ones and give higher scores to the translations picked with most annotators, and penalize the wrong ones taking into account the number of supplied translations. Let item i belongs to the set of instances I belong to a headword. Let T_i is the set of gold translations supplied by annotators for i and S_i is the set of supplied translations by the system, then best score for i :

$$best\ score(i) = \frac{\sum_{s\in S_i} frequency(s \in T_i)}{|S_i| . |T_i|} \quad (3)$$

Precision is calculated by adding the scores and dividing by the number of items attempted by the system, thus penalizing for increasing the number of supplied translations by the system. On the other hand, recall divides by the sum of the scores for each item i by $|I|$.

$$best\ precision = \frac{\sum_i best\ score(i)}{|i \in I : defined(S_i)|} \quad (4)$$

$$best\ recall = \frac{\sum_i best\ score(i)}{|I|} \quad (5)$$

The oot metric allows the systems to supply up to ten translations per item and it does not penalize the system with the number of supplied translations.

$$oot\ score(i) = \frac{\sum_{s\in S_i} frequency(s \in T_i)}{|T_i|} \quad (6)$$

$$oot\ precision = \frac{\sum_i oot\ score(i)}{|i \in I : defined(S_i)|} \quad (7)$$

$$oot\ recall = \frac{\sum_i oot\ score(i)}{|I|} \quad (8)$$

	CBOW	SKIP-G	GloVe
SIM	Euclidean	Manhattan	Manhattan
NORM	False	True	False
MAXO	1	1	1
MINSIM	0	0	0
SEMLAYER	True	True	True
MAXTRNS	2	2	2
AVG	False	False	False
HWINDOW	5	5	ALL
MWINDOW	5	10	10

Table 4: Shows the values of the parameters used by our systems.

Systems	R	P	Mode R	Mode P
UBA-T	27.15	27.15	57.2	57.2
USPWLV	26.81	26.81	58.85	58.85
GloVe	***26.7***	***26.7***	***54.05***	***54.05***
CBOW	***26.35***	***26.35***	***53.36***	***53.36***
SKIP-G	***25.53***	***25.53***	***50.89***	***50.89***
ColSlm	25.99	27.59	56.24	59.16
WLVUSP	25.27	25.27	52.81	52.81
SWAT-E	21.46	21.46	43.21	43.21
UvT-v	21.09	21.09	43.76	43.76
CU-SMT	20.56	21.62	44.58	45.01
UBA-W	19.68	19.68	39.09	39.09
UvT-g	19.59	19.59	41.02	41.02
SWAT-S	18.87	18.87	36.63	36.63
ColEur	18.15	19.47	37.72	40.03
IRST-1	15.38	22.16	33.47	45.95
IRSTbs	13.21	22.51	28.26	45.27
TYO	8.39	8.62	14.95	15.31
DICT	24.34	24.34	50.34	50.34
DICTCORP	15.09	15.09	29.22	29.22

Table 5: Shows the 'best' scores of the systems participating the CLLS, semEval 2010 task 2.

According to these metrics, the theoretical upper bound[5] if all items are attempted and only one translation is supplied: best_{up}=40.57, oot_{up}=405.78

Results and Evaluation

We employed our models in the CLLS task using the configuration parameters in Table 4 and compared our results using the "best" and "oot" measures to other systems competed in the task (Mihalcea et al. 2010) (Table 5, 6). By examining the results, we notice that our systems outperform the state of the art system in the "oot" measure, and keeping a very reasonable performance in the "best" measure. By considering the scores of other systems, we can notice that systems performing well in one measure usually perform poorly in the other measure. For example, if we take "UBA-T" system, it is ranked first in the "best" measure, but ranked tenth in the "oot", the same happens with "SWAT-E" that is ranked first in the "oot", but eighth in the "best". Our systems on the other hand, achieve a considerable balance between the two measures. We examine the effect of using all the possible translations supported by Google translate for the headwords including the infrequent translations (Figure 1, 2, and 3), we can notice that using one translations only can achieve the highest scores. This fact is also proved by the baseline (DICT) which achieve remarkably good scores in the "best" measure. The naïve baseline picks the first translation of the headword and assigns it to all instances, this suggests that the semEval task2 has a data problem, it contains 100 English headwords with 10 instances each. Ideally, these 10 instances per headword should represent distinct senses such that they should not share translations. This is hard to achieve under the restriction of having exactly 10 instances per headword, because not all English words show that much fine-grained polysemous behavior which results in overlapping correct translations between instances, and enables only one translation for all instances to perform well. One challenge CLLS imposes on our techniques, is the need to obtain a list of headword translations that ideally should cover all the gold (reference) translations supplied by annotators. Limited or incomplete list can cause some instances to receive false translations.

Systems	R	P	Mode R	Mode P	dups
GloVe	***267.04***	***267.04***	***54.05***	***54.05***	***1000***
CBOW	***263.54***	***263.54***	***53.36***	***53.36***	***1000***
SKIP-G	***255.26***	***255.26***	***50.89***	***50.89***	***1000***
SWAT-E	174.59	174.59	66.94	66.94	968
SWAT-S	97.98	97.98	79.01	79.01	872
UvT-v	58.91	58.91	62.96	62.96	345
UvT-g	55.29	55.29	73.94	73.94	146
UBA-W	52.75	52.75	83.54	83.54	0
WLVUSP	48.48	48.48	77.91	77.91	64
UBA-T	47.99	47.99	81.07	81.07	-
USPWLV	47.6	47.6	79.84	79.84	30
ColSlm	43.91	46.61	65.98	69.41	509
ColEur	41.72	44.77	67.35	71.47	125
TYO	34.54	35.46	58.02	59.16	-
IRST-1	31.48	33.14	55.42	58.3	-
FCC-LS	23.9	23.9	31.96	31.96	308
IRSTbs	8.33	29.74	19.89	64.44	-
DICT	44.04	44.04	73.53	73.53	30
DITCORP	42.65	42.65	71.6	71.6	-

Table 6: Shows the 'oot' scores of the systems participating the CLLS, semEval 2010 task 2.

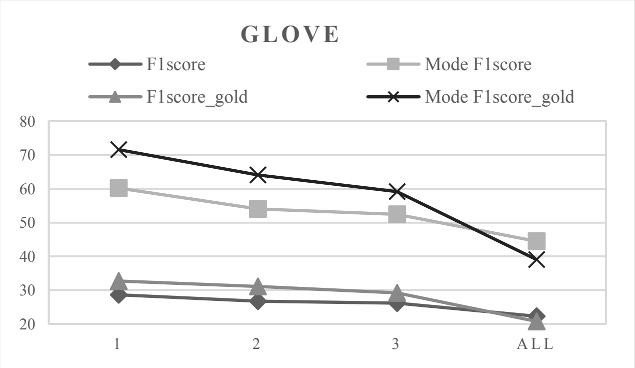

Figure 1: Effect of changing MAXTRNS with/without gold translations on GloVe

[5] The upper bound for both best and oot is multiplied by 100

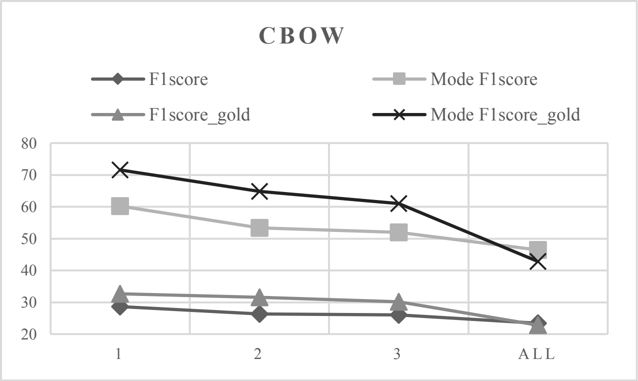

Figure 2: Effect of changing MAXTRNS with/without gold translations on CBOW.

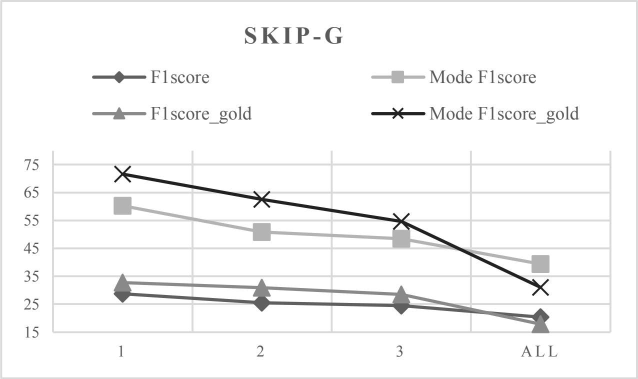

Figure 3: Effect of changing MAXTRNS with/without gold translations on SKIP-GRAM.

Figures 1, 2, and 3 show the performance of our systems with/without using all the possible gold translations for all instances as possible translations for the headword, thus removing the factor of wrong or incomplete headword translations in the evaluation of our systems. Comparing the scores with/without using gold translations; for MAXTRNS=1, 2, and 3, the scores of using the gold translations is better, however at MAXTRNS=ALL, not using the gold translations is better, due to the presence of collocations in the gold translations that are not supported by our systems.

Conclusion and Future Work

We presented a novel technique to solve the CLLS problem that outperformed the state of the art system in the "oot" measure. We introduced the idea of sematic layers using word representation in vector space and showed how it can be effective to capture a concept expressed in a context. We believe our technique explores new grounds in the field of semantic and linguistic computations; it is fast and simple and minimizes the language dependent requirements, which means it is easily applicable to new languages. We would like to address some of the limitations as future work, since translation is a key player in our approach, it will be useful to rely on different sources of translations to ensure the depth and the quality of translations, and find ways to rank these translations. Moreover, handling collocations is essential as many languages show average usage for collocations. Finally, increasing raw data will help building better word vector representations.

References

Mihalcea R.; Sinha R.; and McCarth D. 2010. SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. *In proceedings of the fifth International Workshop on Semantic Evaluation, ACL, Uppsala, Sweden. P*ages 9-14.

Basile P.; and Semeraro G. 2010. UBA: Using Automatic Translation and Wikipedia for Cross-Lingual Lexical Substitution. *In proceedings of the fifth International Workshop on Semantic Evaluation, ACL, Uppsala, Sweden. P*ages 242-247.

Mikolov T.; Sutskever I.; Chen K.; Corrado G.; and Dean J. 2013. Distributed Representation of Words and Phrases and their Compositionality. *In proceedings of Neural Information Processing Systems (NIPS), Nevada, United States.* Pages 3111-3119.

Pennington J.; Socher R.; and Manning C. 2014. Glove: global vectors for word representation. *In the proceeding of the Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar*. Pages 1532-1543**.**

Eisele A.; and Chen Y. 2010. MultiUN: A Multilingual corpus from United Nation Documents. *In proceeding of the International Conference on Language Resources and Evaluation (LREC), May 2010, Valletta, Malta*. Pages 2868-2872.

Skadins R.; Tiedemann J.; Rozis R.; and Deksne D. 2014. Billions of Parallel Words for Free. *In Proceedings of International Conference on Language Resources and Evaluation (LREC), Reykjavik, Iceland.* Pages 1850-1855.

Tiedemann J. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. *In the proceeding on Recent Advances in Natural Language Processing (RANLP), Amsterdam/Philadelphia*. Pages 237-248.

Tiedemann J. 2012. Parallel Data, Tools and Interfaces in OPUS. *In Proceedings of the eighth International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey.* Pages 2214-2218.

Wicentowski R.; Kelly M.; and Lee R. 2010. Swat: Cross-lingual lexical substitution using local context matching, bilingual dictionaries and machine translation. *In Proceedings of the fifth International Workshop on Semantic Evaluation. Association for Computational Linguistics.* Pages 123-128.

Gompel M. 2010. UvT-WSD1: a Cross-Lingual Word Sense Disambiguation system. *In Proceedings of the fifth International Workshop on Semantic Evaluation. Association for Computational Linguistics.* Pages 238-241.

Aziz W.; Specia L. 2010. USPwlv and WLVusp: Combining Dictionaries and Contextual Information for Cross-Lingual Lexical Substitution. *In Proceedings of the fifth International Workshop on Semantic Evaluation. Association for Computational Linguistics.* Pages 117-122.