SWAT: Cross-Lingual Lexical Substitution using Local Context Matching, Bilingual Dictionaries and Machine Translation

Richard Wicentowski, Maria Kelly, Rachel Lee
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081 USA
richardw@cs.swarthmore.edu, {mkelly1,rlee1}@sccs.swarthmore.edu

Abstract

We present two systems that select the most appropriate Spanish substitutes for a marked word in an English test sentence. These systems were official entries to the SemEval-2010 Cross-Lingual Lexical Substitution task. The first system, SWAT-E, finds Spanish substitutions by first finding English substitutions in the English sentence and then translating these substitutions into Spanish using an English-Spanish dictionary. The second system, SWAT-S, translates each English sentence into Spanish and then finds the Spanish substitutions in the Spanish sentence. Both systems exceeded the baseline and all other participating systems by a wide margin using one of the two official scoring metrics.

1 Introduction

We present two systems submitted as official entries to the SemEval-2010 Cross-Lingual Lexical Substitution task (Mihalcea et al., 2010). In this task, participants were asked to substitute a single marked word in an English sentence with the most appropriate Spanish translation(s) given the context. On the surface, our two systems are very similar, performing monolingual lexical substitution and using translation tools and bilingual dictionaries to make the transition from English to Spanish.

2 Scoring

The task organizers used two scoring metrics adapted from the SemEval-2007 English Lexical Substitution task (McCarthy and Navigli, 2007). For each test item i, human annotators provided a multiset of substitutions, T_i, that formed the gold standard. Given a system-provided multiset answer S_i for test item i, the *best* score for a single test item is computed using (1). Systems were allowed to provide an unlimited number of responses in S_i, but each item's *best* score was divided by the number of answers provided in S_i.

$$\textit{best}\text{ score} = \frac{\sum_{s \in S_i} \text{frequency}(s \in T_i)}{|S_i| \cdot |T_i|} \tag{1}$$

The out-of-ten score, henceforth *oot*, limited systems to a maximum of 10 responses for each test item. Unlike the *best* scoring method, the final score for each test item in the *oot* method is not divided by the actual number of responses provided by the system; therefore, systems could maximize their score by always providing exactly 10 responses. In addition, since S_i is a multiset, the 10 responses in S_i need not be unique.

$$\textit{oot}\text{ score} = \frac{\sum_{s \in S_i} \text{frequency}(s \in T_i)}{|T_i|} \tag{2}$$

Further details on the *oot* scoring method and its impact on our systems can be found in Section 3.4.

The final *best* and *oot* score for the system is computed by summing the individual scores for each item and, for recall, dividing by the number of tests items, and for precision, dividing by the number of test items answered. Our systems provided a response to every test item, so precision and recall are the same by this definition.

For both *best* and *oot*, the Mode recall (similarly, Mode precision) measures the system's ability to identify the substitute that was the annotators' most frequently chosen substitute, when such a most frequent substitute existed (McCarthy and Navigli, 2007).

3 Systems

Our two entries were SWAT-E and SWAT-S. Both systems used a two-step process to obtain a ranked list of substitutes. The SWAT-E system first used a monolingual lexical substitution algorithm to provide a ranked list of English substitutes and then these substitutes were translated into Spanish to obtain the cross-lingual result. The SWAT-S system performed these two steps in the reverse order: first, the English sentences were translated into Spanish and then the monolingual lexical substitution algorithm was run on the translated output to provide a ranked list of Spanish substitutes.

3.1 Syntagmatic coherence

The monolingual lexical substitution algorithm used by both systems is an implementation of the syntagmatic coherence criterion used by the IRST2 system (Giuliano et al., 2007) in the SemEval-2007 Lexical Substitution task.

For a sentence H_w containing the target word w, the IRST2 algorithm first compiles a set, E, of candidate substitutes for w from a dictionary, thesaurus, or other lexical resource. For each $e \in E$, H_e is formed by substituting e for w in H_w. Each n-gram ($2 \leq n \leq 5$) of H_e containing the substitute e is assigned a score, f, equal to how frequently the n-gram appeared in a large corpus.

For all triples (e, n, f) where $f > 0$, we add (e, n, f) to E'. E' is then sorted by n, with ties broken by f. The highest ranked item in E', therefore, is the triple containing the synonym e that appeared in the longest, most frequently occurring n-gram. Note that each candidate substitute e can appear multiple times in E': once for each value of n.

The list E' becomes the final output of the syntagmatic coherence criterion, providing a ranking for all candidate substitutes in E.

3.2 The SWAT-E system

3.2.1 Resources

The SWAT-E system used the English Web1T 5-gram corpus (Brants and Franz, 2006), the Spanish section of the Web1T European 5-gram corpus (Brants and Franz, 2009), Roget's online thesaurus[1], NLTK's implementation of the Lancaster Stemmer (Loper and Bird, 2002), Google's online English-Spanish dictionary[2], and SpanishDict's online dictionary[3]. We formed a single Spanish-English dictionary by combining the translations found in both dictionaries.

3.2.2 Ranking substitutes

The first step in the SWAT-E algorithm is to create a ranked list of English substitutes. For each English test sentence H_w containing the target word w, we use the syntagmatic coherence criterion described above to create E', a ranking of the synonyms of w taken from Roget's thesaurus. We use the Lancaster stemmer to ensure that we count all morphologically similar lexical substitutes.

Next, we use a bilingual dictionary to convert our candidate English substitutes into candidate Spanish substitutes, forming a new ranked list S'. For each item (e, n, f) in E', and for each Spanish translation s of e, we add the triple (s, n, f) to S'. Since different English words can have the same Spanish translation s, we can end up with multiple triples in S' that have the same values for s and n. For example, if s_1 is a translation of both e_1 and e_2, and the triples $(e_1, 4, 87)$ and $(e_2, 4, 61)$ appear in E', then S' will contain the triples $(s_1, 4, 87)$ and $(s_1, 4, 61)$. We merge all such "duplicates" by summing their frequencies. In this example, we would replace the two triples containing s_1 with a new triple, $(s_1, 4, 148)$. After merging all duplicates, we re-sort S' by n, breaking ties by f. Notice that since triples are merged only when both s and n are the same, Spanish substitutes can appear multiple times in S': once for each value of n.

At this point, we have a ranked list of candidate Spanish substitutes, S'. From this list S', we keep only those Spanish substitutes that are direct translations of our original word w. The reason for doing this is that some of the translations of the synonyms of w have no overlapping meaning with w. For example, the polysemous English noun "bug" can mean a flaw in a computer program (cf. test item 572). Our thesaurus lists "hitch" as a synonym for this sense of "bug". Of course, "hitch" is also polysemous, and not every translation of "hitch" into Spanish will have a meaning that overlaps with the original "bug" sense. Translations such as "enganche", having the "trailer hitch" sense, are certainly not appropriate substitutes for this, or any, sense of the word "bug". By keeping only those substitutes that are also translations of the original word w, we maintain a cleaner list of candidate substitutes. We call this filtered list of candidates S.

3.2.3 Selecting substitutes

For each English sentence in the test set, we now have a ranked list of cross-lingual lexical substi-

[1] http://thesaurus.com

[2] http://www.google.com/dictionary

[3] http://www.spanishdict.com

```
$best = \{(s_1, n_1, f_1)\}$
$j \leftarrow 2$
while $(n_j == n_1)$ and $(f_j \geq 0.75 * f_1)$ do
    $best \leftarrow best \cup \{(s_j, n_j, f_j)\}$
    $j \leftarrow j + 1$
end while
```

Figure 1: The method for selecting multiple answers in the *best* method used by SWAT-E

tutes, S. In the *oot* scoring method, we selected the top 10 substitutes in the ranked list S. If there were less than 10 items (but at least one item) in S, we duplicated answers from our ranked list until we had made 10 guesses. (See Section 3.4 for further details on this process.) If there were no items in our ranked list, we returned the most frequent translations of w as determined by the unigram counts of these translations in the Spanish Web1T corpus.

For our *best* answer, we returned multiple responses when the highest ranked substitutes had similar frequencies. Since S was formed by transferring the frequency of each English substitute e onto all of its Spanish translations, a single English substitute that had appeared with high frequency would lead to many Spanish substitutes, each with high frequencies. (The frequencies need not be exactly the same due to the merging step described above.) In these cases, we hedged our bet by returning each of these translations.

Representing the i-th item in S as (s_i, n_i, f_i), our procedure for creating the *best* answer can be found in Figure 1. We allow all items from S that have the same value of n as the top ranked item and have a frequency at least 75% that of the most frequent item to be included in the best answer.

Of the 1000 test instances, we provided a single "best" candidate 630 times, two candidates 253 times, three candidates 70 times, four candidates 30 times, and six candidates 17 times. (We never returned five candidates).

3.3 SWAT-S

3.3.1 Resources

The SWAT-S system used both Google's[4] and Yahoo's[5] online translation tools, the Spanish section of the Web1T European 5-gram corpus, Roget's online thesaurus, TreeTagger (Schmid, 1994) for morphological analysis and both Google's and Yahoo's[6] English-Spanish dictionaries. We formed a single Spanish-English dictionary by combining the translations found in both dictionaries.

3.3.2 Ranking substitutes

To find the cross-lingual lexical substitutes for a target word in an English sentence, we first translate the sentence into Spanish and then use the syntagmatic coherence criterion on the translated Spanish sentence.

In order to perform this monolingual Spanish lexical substitution, we need to be able to identify the target word we are attempting to substitute in the translated sentence. We experimented with using Moses (Koehn et al., 2007) to perform the machine translation and produce a word alignment but we found that Google's online translation tool produced better translations than Moses did when trained on the Europarl data we had available.

In the original English sentence, the target word is marked with an XML tag. We had hoped that Google's translation tool would preserve the XML tag around the translated target word, but that was not the case. We also experimented with using quotation marks around the target word instead of the XML tag. The translation tool often preserved quotation marks around the target word, but also yielded a different, and anecdotally worse, translation than the same sentence without the quotation marks. (We will, however, return to this strategy as a backoff method.) Although we did not experiment with using a stand-alone word alignment algorithm to find the target word in the Spanish sentence, Section 4.3 provides insights into the possible performance gains possible by doing so.

Without a word alignment, we were left with the following problem: Given a translated Spanish sentence H, how could we identify the word w that is the translation of the original English target word, v? Our search strategy proceeded as follows.

1. We looked up v in our English-Spanish dictionary and searched H for one of these translations (or a morphological variant), choosing the matching translation as the Spanish target word. If the search yielded multiple matches, we chose the match that was in the most similar position in the sentence to the position of v in

[4] http://translate.google.com/

[5] http://babelfish.yahoo.com/

[6] http://education.yahoo.com/reference/dict_en_es/

the English sentence. This method identified a match in 801 of the 1000 test sentences.

2. If we had not found a match, we translated each word in H back into English, one word at a time. If one of the re-translated words was a synonym of v, we chose that word as the target word. If there were multiple matches, we again used position to choose the target.
3. If we still had no match, we used Yahoo's translation tool instead of Google's, and repeated steps 1. and 2. above.
4. If we still had no match, we reverted to using Google's translation tool, this time explicitly offsetting the English target word with quotation marks.

In 992 of the 1000 test sentences, this four-step procedure produced a Spanish sentence H_w with a target w. For each of these sentences, we produced E', the list of ranked Spanish substitutes using the syntagmatic selection coherence criterion described in Section 3.1. We used the Spanish Web1T corpus as a source of n-gram counts, and we used the Spanish translations of v as the candidate substitution set E. For the remaining 8 test sentences where we could not identify the target word, we set E' equal to the top 10 most frequently occurring Spanish translations of v as determined by the unigram counts of these translations in the Spanish Web1T corpus.

3.3.3 Selecting substitutes

For each English sentence in the test set, we selected the single best item in E' as our answer for the *best* scoring method.

For the *oot* scoring method, we wanted to ensure that the translated target word w, identified in Section 3.3.2, was represented in our output, even if this substitute was poorly ranked in E'. If w appeared in E', then our *oot* answer was simply the first 10 entries in E'. If w was not in E', then our answer was the top 9 entries in E' followed by w.

As we had done with our SWAT-E system, if the *oot* answer contained less than 10 items, we repeated answers until we had made 10 guesses. See the following section for more information.

3.4 *oot* selection details

The metric used to calculate *oot* precision in this task (Mihalcea et al., 2010) favors systems that always propose 10 candidate substitutes over those that propose fewer than 10 substitutes. For each test item the *oot* score is calculated as follows:

$$\textit{oot}\text{ score} = \frac{\sum_{s \in S_i} \text{frequency}(s \in T_i)}{|T_i|}$$

The final *oot* recall is just the average of these scores over all test items. For test item i, S_i is the multiset of candidates provided by the system, T_i is the multiset of responses provided by the annotators, and frequency$(s \in T_i)$ is the number of times each item s appeared in T_i.

Assume that T_i = {*feliz*, *feliz*, *contento*, *alegre*}. A system that produces S_i = {*feliz*, *contento*} would receive a score of $\frac{2+1}{4} = 0.75$. However a system that produces S_i with *feliz* and *contento* each appearing 5 times would receive a score of $\frac{5*2+5*1}{4} = 3.75$. Importantly, a system that produced S_i = {*feliz*, *contento*} plus 8 other responses that were not in the gold standard would receive the same score as the system that produced only S_i = {*feliz*, *contento*}, so there is never a penalty for providing all 10 answers.

For this reason, in both of our systems, we ensure that our *oot* response always contains exactly 10 answers. To do this, we repeatedly append our list of candidates to itself until the length of the list is equal to or exceeds 10, then we truncate the list to exactly 10 answers. For example, if our original candidate list was [a, b, c, d], our final *oot* response would be [a, b, c, d, a, b, c, d, a, b].

Notice that this is not the only way to produce a response with 10 answers. An alternative would be to produce a response containing [a, b, c, d] followed by 6 other unique translations from the English-Spanish dictionary. However, we found that padding the response with unique answers was far less effective than repeating the answers returned by the syntagmatic coherence algorithm.

4 Analysis of Results

Table 1 shows the results of our two systems compared to two baselines, DICT and DICTCORP, and the upper bound for the task.[7] Since all of these systems provide an answer for every test instance, precision and recall are always the same. The upper bound for the *best* metric results from returning a single answer equal to the annotators' most frequent substitute. The upper bound for the *oot* metric is obtained by returning the annotator's most frequent substitute repeated 10 times.

[7] Details on the baselines and the upper bound can be found in (Mihalcea et al., 2010).

	best		*oot*	
System	R	Mode R	R	Mode R
SWAT-E	21.5	43.2	174.6	66.9
SWAT-S	18.9	36.6	98.0	79.0
DICT	24.3	50.3	44.0	73.5
DICTCORP	15.1	29.2	29.2	29.2
upper bound	40.6	100.0	405.9	100.0

Table 1: System performance using the two scoring metrics, *best* and *oot*. All test instances were answered, so precision equals recall. DICT and DICTCORP are the two baselines.

Like the IRST2 system (Giuliano et al., 2007) submitted in the 2007 Lexical Substitution task, our system performed extremely well on the *oot* scoring method while performing no better than average on the *best* method. Further analysis should be done to determine if this is due to a flaw in the approach, or if there are other factors at work.

4.1 Analysis of the *oot* method

Our *oot* performance was certainly helped by the fact that we chose to provide 10 answers for each test item. One way to measure this is to score both of our systems with all duplicate candidates removed. We can see that the recall of both systems drops off sharply: SWAT-E drops from 174.6 to 36.3, and SWAT-S drops from 98.0 to 46.7. As was shown in Section 3.4, the *oot* system should always provide 10 answers; however, 12.8% of the SWAT-S test responses, and only 3.2% of the SWAT-E test responses contained no duplicates. In fact, 38.4% of the SWAT-E responses contained only a single unique answer. Providing duplicate answers allowed us to express confidence in the substitutes found. If duplicates were forbidden, simply filling any remaining answers with other translations taken from the English-Spanish dictionary could only serve to increase performance.

Another way to measure the effect of always providing 10 answers is to modify the responses provided by the other systems so that they, too, always provide 10 answers. Of the 14 submitted systems, only 5 (including our systems) provided 10 answers for each test item. Neither of the two baseline systems, DICT and DICTCORP, provided 10 answers for each test item. Using the algorithm described in Section 3.4, we re-scored each of the systems with answers duplicated so that each response contained exactly 10 substitutes. As shown in Table 2, both systems still far exceed the baseline, SWAT-E remains the top scoring system, and SWAT-S drops to 3rd place behind IRST-1, which had finished 12th with its original submission.

	filled oot		*oot*	
System	R	P	R	P
SWAT-E	174.6	174.6	174.6	174.6
IRST-1	126.0	132.6	31.5	33.1
SWAT-S	98.0	98.0	98.0	98.0
WLVUSP	86.1	86.1	48.5	48.5
DICT	71.1	71.1	44.0	44.0
DICTCORP	66.7	66.7	15.1	15.1

Table 2: System performance using *oot* for the top 4 systems when providing exactly 10 substitutes for all answered test items ("filled oot"), as well as the score as submitted ("oot").

4.2 Analysis of *oot* Mode R

Although the SWAT-E system outperformed the SWAT-S system in *best* recall, *best* Mode recall ("Mode R"), and *oot* recall, the SWAT-S system outperformed the SWAT-E system by a large margin in *oot* Mode R (see Table 1). This result is easily explained by first referring to the method used to compute Mode recall: a score of 1 was given to each test instance where the *oot* response contained the annotators' most frequently chosen substitute; otherwise 0 was given. The average of these scores yields Mode R. A system can maximize its Mode R score by always providing 10 unique answers. SWAT-E provided an average of 3.3 unique answers per test item and SWAT-S provided 6.9 unique answers per test item. By providing more than twice the number of unique answers per test item, it is not at all surprising that SWAT-S outperformed SWAT-E in the Mode R measure.

4.3 Analysis of SWAT-S

In the SWAT-S system, 801 (of 1000) test sentences had a direct translation of the target word present in Google's Spanish translation (identified by step 1 in Section 3.3.2). In these cases, the resulting output was better than those cases where a more indirect approach (steps 2-4) was necessary. The *oot* precision on the test sentences where the target was found directly was 101.3, whereas the precision of the test sentences where a target was found more indirectly was only 84.6. The 8 sentences where the unigram backoff was

	best		*oot*	
SWAT-E	P	Mode P	P	Mode P
adjective	25.94	50.67	192.78	85.78
noun	22.34	40.44	197.87	59.11
verb	18.62	41.46	155.16	55.12
adverb	15.68	33.78	119.51	66.22
SWAT-S	P	Mode P	P	Mode P
adjective	21.70	40.00	126.41	86.67
noun	24.77	45.78	107.85	82.22
verb	13.58	27.80	69.04	71.71
adverb	10.46	22.97	80.26	66.22

Table 3: Precision of *best* and *oot* for both systems, analyzed by part of speech.

used had a precision of 77.4. This analysis indicates that using a word alignment tool on the translated sentence pairs would improve the performance of the method. However, since the precision in those cases where the target word could be identified was only 101.3, using a word alignment tool would almost certainly leave SWAT-S as a distant second to the 174.6 precision achieved by SWAT-E.

4.4 Analysis by part-of-speech

Table 3 shows the performance of both systems broken down by part-of-speech. In the IRST2 system submitted to the 2007 Lexical Substitution task, adverbs were the best performing word class, followed distantly by adjectives, then nouns, and finally verbs. However, in this task, we found that adverbs were the hardest word class to correctly substitute. Further analysis should be done to determine if this is due to the difficulty of the particular words and sentences chosen in this task, the added complexity of performing the lexical substitution across two languages, or some independent factor such as the choice of thesaurus used to form the candidate set of substitutes.

5 Conclusions

We presented two systems that participated in the SemEval-2010 Cross-Lingual Lexical Substitution task. Both systems use a two-step process to obtain the lexical substitutes. SWAT-E first finds English lexical substitutes in the English sentence and then translates these substitutes into Spanish. SWAT-S first translates the English sentences into Spanish and then finds Spanish lexical substitutes using these translations.

The official competition results showed that our two systems performed much better than the other systems on the *oot* scoring method, but that we performed only about average on the *best* scoring method.

The analysis provided here indicates that the *oot* score for SWAT-E would hold even if every system had its answers duplicated in order to ensure 10 answers were provided for each test item. We also we showed that a word alignment tool would likely improve the performance of SWAT-S, but that this improvement would not be enough to surpass SWAT-E.

References

T. Brants and A. Franz. 2006. Web 1T 5-gram, ver. 1. LDC2006T13, Linguistic Data Consortium, Philadelphia.

T. Brants and A. Franz. 2009. Web 1T 5-gram, 10 European Languages, ver. 1. LDC2009T25, Linguistic Data Consortium, Philadelphia.

Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. 2007. FBK-irst: Lexical Substitution Task Exploiting Domain and Syntagmatic Coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007).*

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions.*

E. Loper and S. Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics.*

D. McCarthy and R. Navigli. 2007. SemEval-2007 Task 10: English lexical substitution task. In *Proceedings of SemEval-2007.*

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. Semeval-2010 Task 2: Cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010).*

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