Using the Web as a Linguistic Resource to Automatically Correct Lexico-Syntactic Errors

Matthieu Hermet[1], Alain Désilets[2], Stan Szpakowicz[1,3]
1- University of Ottawa, 2- National Research Council of Canada, 3- Polish Academy of Sciences
{mhermet, szpak}@site.uottawa.ca, alain.desilets@nrc-cnrc.gc.ca

Abstract

This paper presents an algorithm for correcting language errors typical of second-language learners. We focus on preposition errors, which are very common among second-language learners but are not addressed well by current commercial grammar correctors and editing aids. The algorithm takes as input a sentence containing a preposition error (and possibly other errors as well), and outputs the correct preposition for that particular sentence context. We use a two-phase hybrid rule-based and statistical approach. In the first phase, rule-based processing is used to generate a short expression that captures the context of use of the preposition in the input sentence. In the second phase, Web searches are used to evaluate the frequency of this expression, when alternative prepositions are used instead of the original one. We tested this algorithm on a corpus of 133 French sentences written by intermediate second-language learners, and found that it could address 69.9% of those cases. In contrast, we found that the best French grammar and spell checker currently on the market, Antidote, addressed only 3% of those cases. We also showed that performance degrades gracefully when using a corpus of frequent n-grams to evaluate frequencies.

1. Introduction

In this paper, we describe and evaluate an algorithm that leverages the Web as a linguistic resource to automatically correct preposition errors in French texts written by second-language learners. This work is done in the context of Computer-Assisted Language Learning (CALL) — tools to assist with First-Language and Second-Language Acquisition (Davies, 2007).

Most CALL tools available today offer closed deterministic solutions such as fill-in-the-blank drills and multiple-choice questions; see Bax (2003) on the future of CALL. There is, however, a gradual shift towards supporting autonomous learning in more open-ended situations. Many of those scenarios call for systems that can automatically evaluate free-text material produced by the learner, and show them where and how it can be improved. Two types of technologies can be used for this task. Grammar checkers can analyze syntactic correctness, while corpus-based tools can check lexical relationships such as idiomatic noun co-occurences.

Unfortunately, corpora are rarely large enough to sufficiently cover the broad range of lexical patterns present in a given language, which means that some lexical phenomena are left unanalyzed. We believe however that using the whole Web as a linguistic corpus may help deal with that issue. This paper discusses a first attempt at automatically correcting lexico-syntactic errors using the Web as a corpus, and focuses specifically on preposition errors.

Our work is in the vein of a very active recent area of research into the use of the Web as a linguisitic resource (Kilgarriff and Grefenstette, 2003; Cilibrasi and Vitanyi, 2007), including its use in a study of preposition collocations (Isaac et al., 2001).

2. Preposition Choice as an Important Problem for Second-Language Learners

Prepositions sit somewhere between function words (determiners and pronouns) and content words (nouns, verbs, adjectives, adverbs). Like content words, prepositions tend to carry more meaning than function words, and their use tends to evolve with time. But like function words, the list of prepositions is small compared with the overall lexicon. In , for example, there are only 85 simple word prepositions and up to 222 compound prepositions. Also like function words, prepositions tend to occur very frequently in spite of their small number. According to some statistics (Fort and Guillaume, 2007), the percentage of prepositional tokens tends to hover around 14% in most languages.

Prepositions present a particular challenge for second-language learners, because a given first language preposition will often translate to different second language prepositions depending on the context of use. For example, while the English preposition "in" is appropriate for both "I was born *in* Winnipeg" and "I was born *in* America", their French translations use different prepositions ("Je suis né *à* Winnipeg" and "Je suis né *en* Amérique"). The choice of preposition in particular contexts is often idiomatic or depends crucially on lexical semantics. Not surprisingly, the proper choice of preposition is hard to teach, and even harder to validate and correct automatically with software. That is why preposition errors are very common even among intermediary second-language learners. Indeed, we analyzed a corpus of written exercises produced by 30 advanced and intermediary university-level students of French as a second language and found that preposition errors accounted for 17.2% of all errors, and 23.6% of all lexico-syntactic errors.

At the moment, there do not exist good algorithms for automatically identifying and correcting preposition errors. Grammar checkers usually solve the problem of lexical selection through the use of corpus-based material: a collocation is matched against the reference material to check its correctness or admissibility. Such is the case for Antidote <www.druide.com>, currently the best spelling and grammar checker for French, which uses this approach to check collocation of content words (but not prepositions). Unfortunately, few corpora are large enough to cover all of a language's lexical properties. Indeed, current grammar checkers, including Antidote, leave most preposition errors undetected, and when they do detect them, they seldom offer correct suggestions. We have therefore decided to experiment with the Web as a linguistic resource that can be leveraged to automatically correct preposition errors.

3. Algorithm Description

Our base algorithm for correcting preposition errors comprises five steps. A concrete example appears in Figure 1.

- **Step 1 (Prune and Generalize)** Given an input sentence with a selected preposition under study, create a pruned and generalized phrase containing that preposition.
- **Step 2 (Generate Alternative Prepositions)** Generate a minimal list of alternative prepositions, those that can easily be confused with the preposition under study.
- **Step 3 (Generate Alternative Phrases)** Generate a list of alternative phrases as follows. For each of the alternative prepositions, create a variant of the pruned and generalized phrase by replacing the preposition under study by the alternative preposition.
- **Step 4 (Evaluate Frequency of Alternative Phrases)** For each of the alternative phrases, evaluate its frequency: send it to a Web search engine, and note the number of hits.
- **Step 5 (Sort Alternative Phrases by Frequency)** Sort the alternative phrases by their number of hits. The alternative preposition used in the most frequent alternative phrase is the suggested correction.

Input sentence:

Ils ont appelé immédiatement <pour> l'aide.
(They immediately called <for> help.)

Step 1: Prune and Generalize

"appeler pour l'aide"
("call for help")

Step 2: Generate Alternative Prepositions

à, avec, de, depuis, en, jusqu'a, par, pendant, sur

Steps 3-5: Generate, Evaluate and Sort Alternative Phrases

appeler à l'aide: 40800 hits
=> "*à*" is the suggested correction
appeler de l'aide: 543 hits
appeler en aide: 25 hits
appeler pour l'aide: 16 hits
appeler avec l'aide: 14 hits
appeler sur l'aide: 1 hit
all other substitutions have 0 hits

Figure 1: Illustration of the algorithm on a simple example.

Steps 1 and 2 warrant additional explanation.

Step 1 is necessary because even for reasonably short sentences the hit count is often zero, irrelevant of what preposition is substituted. This is due to the fact that, in spite of its size, the Web is still sparse compared to the infinitely large number of possible sentences that can be written in a given language. By pruning and generalizing the input sentences, however, we can get a phrase that retains the context of use of the preposition in the input sentence, while still receiving some hits. This generalization is done by means of controlled lemmatization, using the Xerox Incremental Parser (XIP), which is known to perform well in the presence of language errors in the inputs (Ait-Mokhtar and al., 2001). The input sentence is parsed and then reduced to a minimum: a governing syntactic unit and a governed unit, both needed to preserve the sense and context of use of the preposition in the input sentence.

Because the input sentences are written by second-language learners, they tend to contain many errors which complicate parsing. Consequently, we had to take precautions to ensure that these errors do not affect the end-to-end accuracy of our algorithm.

A first precaution was to parse the input sentence as two separate chunks, namely, words that precede the erroneous preposition, and words that follow and include the preposition. This strategy eliminates parsing errors that might otherwise have been caused by the erroneous preposition, since parsing subcategorization controls work from left to right, based on content word information.

A second precaution was to use only those parts of the XIP analyses which are robust enough to be unaffected by errors in the input sentence. Our generalization strategy is mostly based on the leaves of XIP's parse tree, which amounts to a shallow parsing of noun, adjectival or verbal phrases. In addition, XIP also produces a dependency analysis, which tends to be more vulnerable to errors in the input sentence. Consequently, we only used it to disambiguate certain pruning or generalization decisions, such as the lemmatization of past participle verbs under given auxiliary conditions.

We found those two precautions to be sufficient for limiting the effect of parsing errors, and ensure good end-to-end accuracy of our algorithm in the face of multiple errors in the input sentences. Eventually, however, sentences showing too many errors to pass the parsing step should be corrected following a hierarchy of error importance that casts preposition errors as secondary.

Step 2 was devised to minimize running time by reducing the number of queries sent to the search engine. It focuses on a small set of alternative prepositions likely to be confused with the preposition under study. This set is generated using a multi-level semantic categorization of prepositions (see, for instance, Saint-Dizier, 2007). For example, the following French prepositions can be used to qualify duration, and are therefore considered to be confusable: *"pour"*, *"en"*, *"pendant"*, *"depuis"*. Our algorithm uses 13 such categories, with prepositions typically appearing in more than one category. We also created a list of the most commonly used prepositions (*"de"*, *"à"*, *"sur"*, *"avec"*, *"par"*, *"pour"*) and we consider all prepositions to be confusable with those.

	Antidote	**Naïve**	**No Gen.**	**With Gen.**
Accuracy	3.1%	24.8%	18.8%	69.9%
Avg CPU Time (secs)	0	0	13.2	21.4

Table 1: Accuracy and CPU time for different algorithms and baselines.

	Whole French Web	**n-grams with freq. > 40**	**1/1000th of French Web**
Accuracy	69.9%	59.4%	30.8%

Table 2: Effect of corpus size on accuracy.

4. Results

We evaluated the algorithm on a corpus of 133 sentences collected from intermediate second-language learners. These were sentences that contained at least one preposition error. Note that we did not restrict our choice to sentences that contained only preposition errors. Indeed, many of the sentences contained multiple errors, some of which were not related to preposition choice.

Table 1 shows the accuracy of our algorithm compared to two baselines and one variant. The *Antidote* baseline was obtained by giving each sentence to the Antidote grammar checker. The *Naïve* baseline was obtained by suggesting the most commonly used French preposition (*"de"*) as the correction. In the *No-Generalization* variant of our algorithm we skipped Step 1 (*Prune and Generalize*). Finally, *With-Generalization* corresponds to the full-fledged variant of the algorithm described above.

We see that *Antidote* does very poorly (accuracy: 3.1%), which confirms our intuition that preposition errors cannot be corrected solely through syntactic analysis and lexical analysis of smaller corpora. In fact, even the *Naïve* benchmark does much better (accuracy: 24.8%). An interesting finding is that the *No-Generalization* variant does significantly worse than the *Naïve* approach (accuracy : 18.8%). In contrast, the *With-Generalization* variant performs much better (accuracy: 69.9%), illustrating the need for syntactic pruning and generalization before doing the corpus-based analysis.

The average processing time per correction is fairly high for both variants of the algorithm (13.2 and 21.4 seconds)[1]. We work, however, in the context of a CALL application as opposed to, say, a text editor. In a context where students are writing for the sole purpose of learning how to use prepositions, a 20 seconds wait is probably acceptable, especially if the system processes errors in the background while the student is typing.

Note also that most of the CPU time can be attributed to the

[1] The tests were performed with an Intel Core Duo T2600, 2.16 GHz processor, and 2GB of RAM.

fact that we are querying a remote Web search engine (Yahoo!). Consequently, speed could be greatly increased if we were to use a search engine that resides locally on the machine running the correction algorithm. However, it is currently not practical for end users to keep an index of the whole Web on their machine merely for the purpose of correcting errors in texts. Therefore, an interesting practical question is the extent to which the accuracy of the algorithm is affected by the size of the Web corpus used to evaluate frequencies.

In order to investigate this, we devised a simple downscaling scheme to simulate the effect of using a smaller corpus. All frequencies are downscaled by a constant factor, and any frequency whose downscaled frequency is smaller than 1 is deemed to not have occurred in the smaller corpus (in other words, its frequency is rounded down to zero). Using this simple technique, we were able to simulate a situation where frequencies are evaluated based on a database of frequent n-grams like the corpus recently published for English by Google (GoogleResearch, 2006). We were also able to simulate a situation where frequencies are evaluated based on a corpus whose size is one thousandth of the size of the French Web.

Table 2 summarizes the results of this analysis. We see that estimating frequencies based on a corpus of n-grams whose frequency on the French Web exceeds 40 would result in a relatively small decrease in accuracy (69.9% down to 59.4%), and would still leave us with a system that performs significantly better than either the *Naïve* or the *Antidote* benchmark. This is important, since it means that our algorithm could perform well using a linguistic resource whose compressed size is in the order of 24G (the size of the Google English n-gram corpus). On the other hand, we see that estimating frequencies by searching a corpus equivalent to one thousandth of the French Web might significantly decrease accuracy (69.9% down to 30.8%) and result in a system that performs only marginally better than the *Naïve* benchmark.

5. Conclusions and Future Work

We have presented an algorithm that outperforms any known alternative to automatic correction of preposition errors, an instance of lexico-syntactic errors. It is done in the context of Second-Language Learning, specifically learning French. The methodology combines aspects of Natural Language Processing – syntactic parsing and pruning – with simple corpus statistics, namely Web hit counts. This simple algorithm yields a 69.9% accuracy. We have found these initial results encouraging enough to motivate further work. We can see three axes for future research.

A first axis is to consider how the preposition correction algorithm could be improved. For example, we have found that the *Prune and Generalize* step is crucial, but at the moment we only use very basic grammatical analysis patterns to do this. We plan to investigate how more sophisticated syntactic analysis might improve accuracy. We also plan to investigate the effectiveness of pruning strategies which are not language-dependent and might be less vulnerable to errors in the input sentence. For example, one might select n words before and after the preposition under study, and removing words like adjectives and adverbs. Another improvement would be to try and decrease CPU time by using an actual corpus of French n-grams whose frequency on the Web is at least 40.

A second axis would be to develop similar algorithms to solve other types of lexically dependent errors commonly made by second-language learners.

A third axis is to investigate the use of this algorithm in an actual CALL setting. For example, one could develop an interactive system where students write free-form text with the aim of learning about a specific type of errors (for example, choice of preposition). The system would automatically correct errors of that type and provide justification in the form of relevant examples mined from the Web. One could do a controlled experiment to evaluate whether the system actually improves the student's ability to use preposition correctly in future free-form texts they write.

References

Antidote: www.druide.com/antidote.html

S. Ait-Mokhtar, J.-P. Chanod, C. Roux (2001) *A Multi-Input Dependency Parser*. In Proceedings of the Seventh IWPT (International Workshop on Parsing Technologies), Beijing.

S. Bax (2003), "CALL - past, present and future", System 31(1), 13-28

R. L. Cilibrasi, P. M. B. Vitanyi (2007), "The Google Similarity Distance", IEEE Trans. Knowledge and Data Engineering, 19(3), 370-383.

G. Davies (2007), "Computer Assisted Language Learning: Where are we now and where are we going?". Keynote paper originally presented at the UCALL Conference, University of Ulster, Coleraine in June 2005, revised in March 2007.

K. Fort, B. Guillaume (2007), "PrepLex: un lexique des prépositions du français pour l'analyse syntaxique". TALN 2007, Toulouse, June 5-8.

Google n-grams (2006) :

googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

F. Isaac, T. Hamon, C. Fouquere, L. Bouchard, L. Emirkanian (2001), "Extraction informatique de donnees sur le web", Revue DistanceS 5(2), 195-210.

A. Kilgarriff, G. Grefenstette (eds.) (2003), Special Issue on the Web as Corpus. Computational Linguistics, 29(3).

P. Saint-Dizier (2007) "Regroupement des Prepositions par sens". Undated report, IRIT Toulouse. http://www.irit.fr/recherches/ILPL/Site-Equipe/publi_fichier/prepLCS.doc