Algorithm qualifies for C1 courses in German exam without previous knowledge of the language: An example of how corpus linguistics can be a new paradigm in Artificial Intelligence

Rogelio Nazar
Institute for Applied Linguistics
Pompeu Fabra University, Barcelona

Abstract

This paper reports on the results of an experiment in which a computer program was able to obtain a high qualification in a German language multiple choice exam using corpus statistics as the only source of information. The system does not use any kind of explicit knowledge of German grammar nor vocabulary: answers are found by simply querying a search engine and selecting the most frequent combination of words. The result obtained with this experiment is the C1 qualification, which, according to the "Common European Framework of Reference for Languages"[1], is equivalent to the level of a person who is in full command of the language. The general purpose of this proposal is to show that, in some cases, poor-knowledge approaches based on relatively simple statistics can perform better than fully informed symbolic rule-based systems.

1. Introduction

This paper reports on an experiment that originated in the development of statistical techniques for the study of word co-occurrences for a different project[2], one which is aimed at the diachronic study of Spanish, more specifically, at the detection of change in lexical combinatorics. These techniques draw the attention of colleagues working on Computer Assisted Language Learning (Alonso Ramos et al., 2011; Ferraro et al. 2011 and third parties from the private sector) who are developing computer programs for the evaluation of texts produced by a second language learner with the purpose of tracking down eventual errors and suggesting corrections. These are not only spelling corrections but also grammar and even the proper use of collocations, the most difficult part of language learning. At the same time, publishers of second language learning materials are trying to develop programs to automatically generate multiple choice exercises and, of course, to evaluate the result. It is evident that manually

[1] The Council of Europe. http://www.coe.int/t/dg4/linguistic/Source/Framework_en.pdf [Accessed September 2011].

[2] Project APLE: "Updating processes of the Spanish lexicon from the press" Spanish Ministry of Science and Innovation: Ref. FFI2009-12188-C05-01, subprogram FILO. Period: 2010-2012. Project Leader: Dr. M. Teresa Cabré Castellví.

generating multiple choice exercises is a labor and time-consuming task, therefore, it is likely that there would be a high demand for software applications that could produce this kind of exams on the fly[3].

The suggested approach for both tasks, i.e., to evaluate the text produced by the student or to generate multiple choice exercises, was based on a statistical analysis of the web. As a first experiment to test the feasibility of this technique, it was decided that for a software to be able to help students learn a language, it should prove able to pass a language exam itself.

The idea to solve the exam is simply to select the word combination that shows more hits on the web. This strategy has no point of comparison with the complexity and the amount of information needed for achieving a similar result with symbolic and linguistically informed rule-based systems. Naturally, one has to bear in mind that there is a considerable gap between solving multiple choice exams and a real language exam where students also have to write essays and solve speech and reading comprehension tests. In fact, the qualification obtained in this experiment, with one of Goethe Institute's[4] online exams (see Section 3), serves merely as an orientation and does not represent an official qualification by this organization. Regardless of the official status of the qualification, what is really important is that the software was able to answer most of the questions correctly, a performance comparable to someone who is in full command of the language.

2. Related Work

It would not be accurate to attest that statistics or knowledge poor approaches are new in artificial intelligence (AI), let alone that they represent a "paradigm shift". Indeed, the approach presented in this paper could be classified on one of the two paradigms that already exist in AI (Russell & Norvig, 1995) and in other fields such as computational linguistics (Jurafsky & Martin, 2000). One of the paradigms would be the "symbolic approach", the one that gave birth to the so called "expert systems", which are characterized by having large amounts of information about the particular task that the system is aimed to solve. In the case of an automaton whose task is to solve a multiple choice German exam, an expert system would require an extraordinary amount of information about German vocabulary and grammar as well as common sense knowledge (in the form of encyclopedic or factual knowledge) to deal with the information contained in the sentences of the exam. There is no need to say that such an approach would be very expensive to build and maintain and, in the case of this particular task, it would be highly unlikely that the system would perform well, given the intrinsic difficulties of the task at hand.

The other paradigm in AI, the "knowledge poor approach", came at a later period in the history of the field and has shown a steady progress since the late eighties both in AI and in computational linguistics. The main difference with the symbolic

[3] Naturally, among other resources, such a project would entail a taxonomy of language learner errors, such that the exercises are oriented to certain types of difficulties.

[4] The Goethe Institute is a German organization for the education in German language and culture which has a world-wide network of schools. http://www.goethe.de/ [Accessed September 2011].

approach is that instead of having large amounts of information hard-coded into the system, the algorithms of this second trend try to obtain the information directly from the analyzed data by statistical means. This second paradigm is the one adopted in this experiment. What is truly new in the field of AI is to use corpus linguistics as a means for solving what before would have been a typical AI task. Only a few decades ago, the research community did not have the enormous amounts of texts in digital format that are available today. The paradigm shift described here is what some authors call "the Google-scale approach to AI-hard problems" (Robert Dale, personal communication), and it is in this sense that much more dialogue and collaboration between AI and corpus linguistics communities is to be expected in the near future.

As far as the particular task of automatically solving multiple choice language exams is concerned, it is hard to find a direct precedent in the literature. The most related area of research would be the development of software for automated writing assistance, that is, the family of algorithms to check spelling, grammar and style. In principle, any of these devices could be used in an attempt to solve a multiple choice language exam, since ideally the system should only approve one correct option.

In the particular field of error checking, Atwell (1987) is among the first to use a statistical and knowledge poor approach to detect grammatical errors in a field clearly dominated by rule-based approaches (Heidorn et al, 1982; Richardson & Braden-Harder, 1988; Schneider & McCoy, 1998). Atwell used a POS-tagger to detect unlikely POS-tag sequences as error candidates. In the specific task of article determination, rule based approaches (Murata & Nagao, 1993) where outperformed by statistical approaches (Knight & Chandler, 1994; Han et al., 2006). In general grammar checking, however, rule-based approaches are still an active field of work. Language Tools (Naber, 2003), for instance, is a collaborative approach to generate rules for grammar error detection in the open source community. However, the extent to which this kind of efforts can cope with the variety of errors that can be found in a text is still to be determined, especially in texts produced by second language learners. Evidently, statistical analyses will always be able to process more data and are expected to achieve better recall than hand-crafted resources, no matter how many individuals contribute in a rule-based collaborative effort, how well organized they are, or how fine graded their taxonomy of errors is.

Many authors have attempted to combine statistical learning from a large corpus of native data and rule-based techniques. This is the case of Gamon et al. (2008), who designed a system for Chinese and Japanese learners of English. Their system includes web searches as an aid for the user to obtain examples. Chen (2009) reports on a system that outperforms Gamon et al.'s model with a statistical approach using the British National Corpus (BNC) as a reference corpus for word combination. According to Chen, this system also uses search engines but, again, only to let the user browse the web to obtain examples or spelling suggestions. Wible & Tsao (2010) have conducted statistical analyses of the BNC with a technique they call "hybrid n-grams", which are n-grams not only of words but of different layers of information, such as lemmatization and POS-tagging. This approach has a strong potential for generalization from a limited set of data, given that, from a sample of n-grams such as the BNC, they can match a broader number of n-grams produced by learners that are not contained in the BNC. These n-grams are not only practical to test the frequency of use of a given expression,

but also to obtain suggestions from the corpus by using wildcard characters in the n-grams, such as *to * a loan* (something that users of search engines are already doing manually).

In recent years, a large number of reports have been published on error checking using the web as a corpus. These studies share the same core idea: if a particular combination of words is not frequent on the web, this can be taken as indication that something may be wrong with the expression, particularly when the individual words are frequent on their own. The idea that infrequent constructions may not be correct in some language is of course unpleasant for those who consider that originality in written expression is something that must be preserved. In this sense, this kind of technologies would have the undesirable effect of "educating" people towards uniformity, hampering individual style and originality. Another more conciliatory point of view would be to consider that grammar checking technologies are aimed at the first steps of language learning. It is left for a further stage, when students are autonomous enough to continue learning the language by reading and interaction, that they will be able to develop their own ability to produce original prose.

Most authors using the web as a corpus accept an input text and submit this sample to different degrees of processing, at least lemmatization, POS-tagging and chunking. The purpose of this preprocessing is to use only chunks from the input and not the entire sentence as queries for the web search engine, given that long queries tend to match fewer documents. The chunks are mainly combinations that match predefined syntactic patterns (Moré et al, 2004; Yin et al., 2008). Whitelaw et al. (2009) report a strategy somewhat similar to the present paper in that it does not use explicit linguistic knowledge, however they focus in spelling error correction. They evaluate the orthography by querying a search engine with token n-grams from the input text. Sjöbergh (2009) has attempted a similar approach for grammar error correction in Swedish, using the frequency of token bigrams in the web to determine if they are normative uses, but his experiments show that this approach performs significantly worse than rule-based commercial packages for grammar correction in Swedish. The relatively low-density of this language on the web might be among the possible explanations for the poor performance of Sjöbergh's test, however his work suggests that both approaches can complement each other. Hermet et al. (2008), working on prepositional errors in French, report on the use of statistical analysis of the web within a hybrid model. Instead of using word combinations as they appear in text, the authors state that a preprocessing of the input text is fundamental to increase the number of matches on the web. This preprocessing includes lemmatization and parsing in order to reduce the input text to minimum structures of governing and governed units. The sentence is thus separated in chunks before and after the analyzed preposition. For the particular case of preposition errors, they report significant improvement over rule-based commercial grammar checkers (70% accuracy vs. 3%).

All in all, the analysis of related work reveals that the problem of error checking and correction is of great complexity and that there is still much room for improvement. Authors report on a great variety of problems (e.g., preposition vs. collocation) which are so different that cannot be directly compared. The strategies proposed are, as well, very diverse. The present proposal departs from the strategies reported in the literature in the following aspects: 1) in the nature of the experiment (to try to solve a language

exam designed for humans); 2) the fact that it does not target any particular kind of error (and therefore disregards taxonomies of errors); 3) it represents a radically simple method and 4) a truly language independent approach.

3. Methods

The proposed experiment needs only two essential pieces of information to be conducted: 1) a multiple choice exam –that is, exercises in which the user has to select one from a number of given options to fill in a blank space in a sentence– and 2) a large textual corpus. The experiment could be replicated in different languages, provided that they are represented on the web or on other digital corpus. In this case, the experiment is in German, and the web is used as the corpus. German was used for no particular reason apart from not being English. The reason for excluding English is simply that there is already too much data in this language on the web and therefore it would be far more interesting to test the intended strategy in a mid-level density language such as German. In this respect, many other languages are, to a greater or lesser extent, in a similar situation.

The language test for the experiment is the one provided by the Goethe Institute (see footnote 4). In the website of the institute, one can find a free online German exam called "Test your German" (Figure 1), designed for people who want to know which level of a German course they will best fit in. The exam comprises 30 multiple choice questions. The questions range from very basic grammar knowledge, such as selecting the correct preposition or the correct number and/or gender agreement, to the most difficult part of language learning which is to select the correct collocations such as verb-noun or adjective-noun combinations.

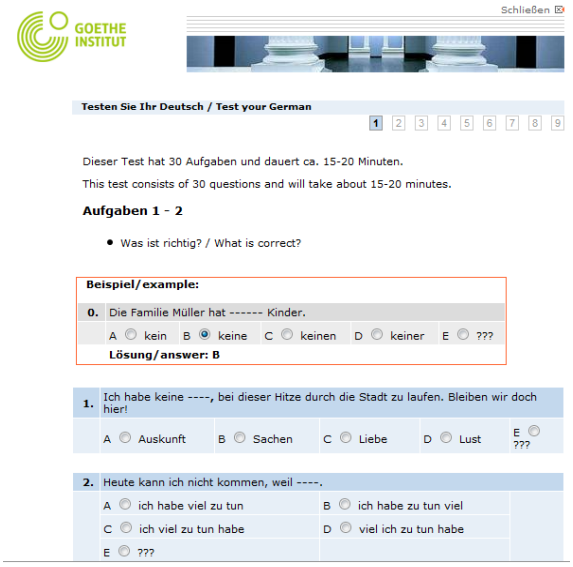

Figure 1: Screenshot of the initial page of the German test

The basic idea to make a computer program solve a language exam like this without any kind of previous knowledge of the language is to use the web as a corpus. The program is able to select the correct answers in most of the cases by querying a

regular search engine (Yahoo Boss Api[5]) with segments of the example sentence given in the exam, each time filling in the blank with each of the different options. Naturally, the idea is that the query that results in the largest number of hits retrieved by the search engine is selected as the correct answer.

For illustration, consider the example in Table 1, provided also in Goethe Institute's exam. The correct answer in this case is B (*Die Familie Müller hat keine Kinder*). We can know that without knowing German by simply querying the search engine with the different possibilities to select the one that yields more results. Of course, it is unlikely to find results using the whole sentence as a query, but it suffices to try with one (or more) word(s) at each side of the blank. In the case of the above example, the different queries would be as follows: *hat kein Kinder*, *hat keine Kinder*, *hat keinen Kinder*, and *hat keiner Kinder*.

```
================================================
0       Die Familie Müller hat * Kinder.
        [A]     kein
        [B]     keine
        [C]     keinen
        [D]     keiner

0        hat * Kinder
        A        hat kein Kinder
                        30
        B        hat keine Kinder
                        5805
        C        hat keinen Kinder
                        7
        D        hat keiner Kinder
                        40

ANSWER: B
================================================
```

Table 1: Example of the solution of an exercise based on hits returned by Yahoo

4. Results

After running the program with the test published on the Goethe Institute web site and submitting the answers back to the test, the score is computed and printed, as shown in Figure 2. The result in this case is that the software answered correctly 21 out of 30 questions; a level that, according to the test, is equivalent to the C1 qualification[6].

[5] The Yahoo Boss Api can be accessed at http://developer.yahoo.com/search/boss/ [September 2011].

[6] The explicit mention of the qualification in terms of the Common European Framework of Reference for Languages (“B1”, “B2”, “C1”, etc) appeared when this experiment was conducted (January 2010). After that, probably for legal reasons, the Goethe Institute removed that qualification and now it only specifies the number of correct answers and some encouraging expressions such as “Prima!” when the user is doing well.

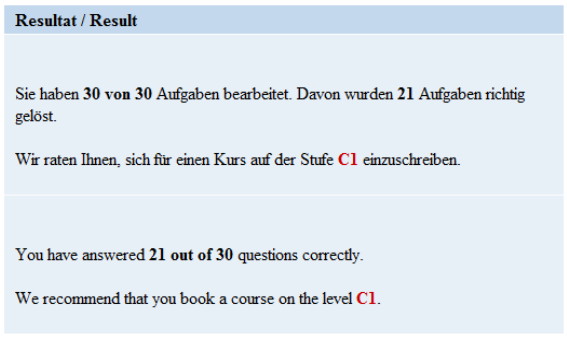

Figure 2: Screenshot of the page showing the score obtained in the test

```
================================================
9       Kommen Sie nicht zu spät. * Sie Ihren Wecker so,
dass Sie nicht nur pünktlich in der Firma sind, sondern auch
genügend Zeit haben, Ihr zukünftiges Büro zu finden.
        [A]     Stell
        [B]     Stelle
        [C]     Stellen

9        spät. * Sie
        A        spät. Stell Sie
                        1
        B        spät. Stelle Sie
                        0
        C        spät. Stellen Sie
                        30

ANSWER: C
================================================
```

Table 2: Example of a solution for exercise number 9

When analyzing the results, the most notable aspect is the difference in the number of hits retrieved for each option in the different exercises. This difference, however, has little impact on the accuracy of the answers, as shown by the results. For instance, in the case of exercise number 9 (Table 2), the decision of the best word combination (option C) has to be taken on the basis of 1 to 30 hits on the search engine, while in the case of exercise 15 (Table 3), the selected option (A) has 835,352 hits.

```
================================================

15      Und nach 48 Stunden haben wir nur * ein Viertel des
Gesprächs im Kopf.
        [A]     noch
        [B]     über
        [C]     weniger


15       nur * ein
        A        nur noch ein
                        835352
        B        nur über ein
                        56530
        C        nur weniger ein
                        78

ANSWER: A

================================================
```

Table 3: Example of a solution for exercise number 15

With respect to error analysis, there are three cases where none of the options matched any documents on the web, as in the example shown in Table 4. In these cases, the presence of relatively rare lexical units in the queries hampers the retrieval of contexts. This is an issue that should be refined in future work. If the queries were subjected to a pre-process where infrequent lexical units were replaced by wildcard

characters, then there would be a better chance for retrieving web data. The remaining errors in the exam were produced mainly for the same reason, in cases where there was not enough data to decide. In other cases, even when there is plenty of data for the particular searches, frequency counts of right choices compete with wrong ones because they produce a combination that matches documents in the web. The example shown in Table 5 is interesting because the selected option ("sonst hätte ich") forms a grammatical and very frequent construction, but inappropriate for this specific context.

```
==========================================
22      Der kleine Junge hatte lange Haare, * ihn viele für
ein Mädchen hielten.
        [A]     darum
        [B]     deshalb
        [C]     denn
        [D]     weshalb

22       Haare, * ihn
        A        Haare, darum ihn
                        0
        B        Haare, deshalb ihn
                        0
        C        Haare, denn ihn
                        0
        D        Haare, weshalb ihn
                        0

ANSWER: A
==========================================
```

Table 4: Example of a case where the algorithm cannot tell the answer because neither of the option matches any documents on the web

```
==========================================
17      Leider waren die Ferien schon zu Ende, sonst * ich
mit den Kindern länger geblieben.
        [A]     hätte
        [B]     wäre
        [C]     werde
        [D]     würde


17      , sonst * ich
        A       , sonst hätte ich
                        152073
        B       , sonst wäre ich
                        46811
        C       , sonst werde ich
                        21230
        D       , sonst würde ich
                        86321

ANSWER: A
==========================================
```

Table 5: An example where a wrong option is more frequent than the correct one (which should be "B")

In order to offer some reference of comparison with other authors work, three rule based grammar checkers were used to try to solve the same German exam: the open source Language Tools (Naber, 2003), the commercial package Grammatica (ULTRALINGUA, 2011) and Microsoft Word. The procedure for using a grammar checker to solve a multiple choice exam is rather simple. For each multiple choice exercise in the exam, different sentences are printed for each of the available options, as shown in Table 6. Ideally, a grammar checker would analyze them and flag sentences A, C, and D as ungrammatical. If this is the case, it is considered that the grammar checker selects option B as the correct answer. In case more than one sentence go unflagged, then one of them is randomly selected. If all sentences of an exercise pass the grammar

checker without being flagged (or if they are all flagged as wrong), then no option is selected (the “don't know” checkbox is marked).

```
============================================
[A]     Die Familie Müller hat kein Kinder.
[B]     Die Familie Müller hat keine Kinder.
[C]     Die Familie Müller hat keinen Kinder.
[D]     Die Familie Müller hat keiner Kinder.
============================================
```

Table 6: Examples of sentences submitted to the grammar checkers

System	**Correct answers (out of 30)**
Random	4
Grammatica	3
Language Tools	3
Microsoft Word	8
This experiment	21

Table 7: Comparison of the results obtained by the proposed algorithm with three grammar checkers and a random baseline

A random selection of the options of the test counts as a final baseline. As we can see in Table 7, the difference in performance between the algorithm described in this paper and the evaluated rule based proposals is highly significant. The reason why the random selection performs slightly better than some checkers is probably because the random baseline always selects one option while the grammar checkers do not select any option if they do not find errors in the sentences.

5. Conclusions and Future Work

This paper has presented a method to evaluate the correctness of a given linguistic expression on the basis of its frequency in a large corpus (i.e. the web). The technique is radically simple in comparison with related approaches and it does not take into account external resources such as explicit linguistic knowledge. Despite its simplicity, the experiment is interesting both from a theoretical and a practical point of view. From a theoretical perspective, this kind of corpus statistics experiment suggests some clues not only about a contextual definition of word combinations, but also on how humans learn and store word combinations (precisely, by some sort of natural “corpus statistics”), though obviously this paper does not constitute evidence in support of that claim. From the practical point of view, in turn, there are different lines of application. As regards the production of second language learning materials, it is important to be able to solve exams, but more important would be to automatically generate multiple choice exercises. This issue will be analyzed in future work, but the basic lines of reasoning have been presented in this paper.

Another possibility of practical interest that remained unexplored in this paper, and will also have to be addressed in the future, is the fact that the options (from the multiple choice test) are not strictly necessary for the solution, since most search engines support the insertion of wildcard characters (*) in the query expressions. Thus, one can compose a query such as “hat * Kinder” or “Müller hat * Kinder”, etc. The task in this case would be to find the most frequent element in the position of the wildcard character in the snippets returned by the search engine. Of course, this would let to competing possibilities (“Müller hat [zwei, drei, vier, etc.] Kinder”) and a certain probability of error, and thus a deeper analysis of the text and a series of filters would be necessary. However, at any rate it is a very promising line of research. The already mentioned work of Wible & Tsao's (2010) on the BNC adopts a similar approach.

Last but not least, potential users of error checking systems are not only beginner language learners. Writing scientific or technical papers in English can be a difficult problem for authors who are not native speakers of English, particularly in the case of conference proceedings, where there is no post-editing or proofreading of the papers. Interestingly, this situation posses a potential problem for an approach such as the one of the present paper: the web is already populated with incorrect English sentences, and this is especially the case in scientific and technical literature for the above mentioned reasons. This is a major problem for a statistical approach that takes the web as a reference corpus. The solution in this case would possibly reside in not using the web but an ad-hoc corpus instead; a corpus constituted by edited journal papers only, where a lower error rate is to be expected. Another factor that mitigates the problem of noise in the web is that the correct version of a text (or its intended meaning) will always be more frequent than the erroneous formulations given that the latter are much more diverse. In any case, one can only hope that all the effort in automated text assistance will not go in detriment of vocabulary richness and originality of expression.

Acknowledgments

This work has been possible thanks to resources generated from Project APLE (see footnote 1). Leo Wanner is to be credited for introducing the author into the field of Computer Assisted Language Learning. I would also like to express my gratitude to Robert Dale, who helped with comments and many references to related work. Owen Johnson also offered interesting ideas for future work. Two native speakers of German (Alexandra Spalek and Sebastian Cramer) provided valuable feedback on the results of the experiment.

References

Alonso Ramos, M; Wanner, L.; Vincze, O.; Nazar, R.; Ferraro, G.; Mosqueira, E., Prieto, S. (2011). Annotation of collocations in a learner corpus for building a learning environment. Proceedings of Learner Corpus Research 2011.

Atwell, E. (1987). How to detect grammatical errors in a text without parsing it. Proceedings of the Third Conference of the European Association for Computational Linguistics, Copenhagen, Denmark, pp. 38–45.

Chen, H. (2009). Evaluating Two Web-based Grammar Checkers: Microsoft ESL Assistant and NTNU Statistical Grammar Checker. Computational Linguistics and Chinese Language Processing, Vol. 14, No. 2, pp. 161–180.

Ferraro, G.; Nazar, R.; Wanner, L. (2011). Collocations: A Challenge in Computer-Assisted Language Learning. Proceedings of the 5th International Conference on Meaning-Text Theory. Barcelona, September 8-9, 2011.

Gamon, M.; Leacock, C.; Brockett, C.; Dolan, W.; Gao, J.; Belenko, D. (2009). Using Statistical Techniques and Web Search to Correct ESL Errors. CALICO Journal, Vol. 26, No 3, pp. 491–511.

Han, N.; Chodorow, M.; Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. Natural Language Engineering, Vol. 12, No. 2, pp. 115–129.

Heidorn, G., Jensen, K. Miller, L; Byrd, R; Chodorow, M. (1982). The EPISTLE text-critiquing system. IBM Systems Journal, No. 21, pp. 305–326.

Hermet, M.; Désilets, A.; Szpakowicz, S. (2008). Using the web as a linguistic resource to automatically correct lexico-syntactic errors. In Proceedings of the Sixth International Conference on Language Resources and Evaluation, Marrekech, Morocco, pp. 390–396.

Knight, K.; Chander, I. (1994). Automated Postediting of Documents. Proceedings of National Conference on Artificial Intelligence, Seattle, USA, pp. 779–784.

Naber, D. (2003). A Rule-Based Style and Grammar Checker. Diplomarbeit Technische Fakultät, Universität Bielefeld.

Moré, J.; Climent, S.; Oliver, A. (2004). A Grammar and Style Checker Based on Internet Searches. Proceedings of LREC 2004, Lisbon, Portugal.

Murata, M; Nagao, M. (1993). Determination of referential property and number of nouns in Japanese sentences for machine translation into English. Proceedings of the 5th TMI, pp. 218–225.

Russell, S.; Norvig, P. (1995). Artificial intelligence: a modern approach. Englewood Cliffs, N.J. Prentice Hall, cop.

Richardson, S. & Braden-Harder, L. (1988). The experience of developing a large-scale natural language text processing system: CRITIQUE. Proceedings of the second conference on Applied natural language processing (ANLC '88). Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 195–202.

Schneider, D.; McCoy, K. (1998). Recognizing syntactic errors in the writing of second language learners. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Montreal, Canada, pp. 1198–1204.

Sjöbergh, J. (2009). The Internet as a Normative Corpus: Grammar Checking with a Search Engine. Techical Report, Dept. of Theoretical Computer Science, Kungliga Tekniska högskolan.

ULTRALINGUA, inc. (2011). “German Grammatica Spelling and Grammar Checker”. http://www.ultralingua.com/ [Accessed September 2011].

Whitelaw, C.; Hutchinson, B.; Chung, G.; Ellis, G. (2009). Using the Web for language independent spellchecking and autocorrection. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 890–899.

Wible, D.; Tsao, N. (2010). Stringnet as a computational resource for discovering and investigating linguistic constructions. Proceedings of the NAACL-HLT Workshop on Extracting and Using Constructions in Computational Linguistics, Los Angeles.

Yin, X; Gao, J.; Dolan, W. (2008). A Web-based English Proofing System for English as a Second Language Users. Proceedings of the 3rd International Joint Conference on Natural Language Processing, Hyderabad, India, pp. 619–624.