Problems of search engine in "phraseology" linguistic software project

Marjan Krašna, Bojan Bedrač, Vida Jesenšek
University of Maribor, Faculty of Arts, Koroška cesta 160, 2000 Maribor
marjan.krasna@um.si, bojan.bedrac@um.si, vida.jesensek@um.si

Abstract: *In the course of a few years automatic language translation has gain huge impact in social interaction. Automatically translated text is far from perfect but still useful in some occasion (especially web browsing). From the analysis of the students' performance (translation studies courses) we have discovered that phrases are hard to translate. In our project (phraseology of German language) we were to develop software support for phrase linguistic analysis and translation. From the preparation of the requirement analysis it becomes obvious that data structure and processing are not the real problem but search engine proves to be much more complex. Searching in the text may be regarded as simple task. In the linguistic analysis a whole range of requirements for text search emerge (language specifics, vocal prediction, delimiters, lemmatization …). A metrics based on the search results and statistics for ranking the displayed results was needed. In the article we explain the consideration of the text search engine and our approach to solve the situations that were detected or could be detected in our project. Some consideration were analyzed but eventually dismissed since they can be resolved with the appropriate user manual.*

Keywords: information system, software development, software service, linguistic, text search

1 Introduction

Communication in foreign languages is one of the eight key competences declared in EU competence framework [1]. Regarding the multicultural and multi-language nature of EU it is necessary for people to know more than one language. Educational systems incorporate learning of foreign languages in different levels of education where students come into contact with foreign language in primary schools. In secondary schools one or two foreign languages are mandatory. Language competence is not something that could be uploaded therefore study courses for different levels of comprehension are required. But the language competence is not something permanent in the mind. It degrades as time goes by especially when not used regularly.

The Faculty of Arts at the University of Maribor has two distinct categories of study programs that correspond to language learning. Students can enroll into translation or didactics language study programs. Study programs cover English, German and Hungarian languages. Translation study programs graduates have competences in translation and interpretation of languages whereas didactics study programs becomes teachers of languages at different levels of education.

In today's world information support for study courses are necessary. Students demand contemporary services. Information support in this matter enables them to get required feedback faster and be more productive.

In the course Electronic translation tools and information systems students acquire knowledge about using ICT in translation. Different aspects of translation services are studied and checked in the computer classroom.

Constant changes and availability of ICT changed the learning procedures. Our language course students rarely use printed dictionaries. Most of their study assignments are similar to the work of translators and is done with the help of a computer. A computer does not only correct spelling errors but becomes practical even in grammatical and translation suggestions. Software is available that helps translating between languages (e.g. Trados, WordFast … and even Microsoft Word) and effectiveness of translators has increased manifolds [2]. But machine translation still has many flaws it is not successful in translation language specific situations (e.g. jokes and phrases). Literal translation usually lost the original meaning despite the fact that it may be understandable. It was long considered that a good translator or language learner should learn the phrases and use them. There are some phrases that are frequently used and most people know them and phrases that are rarely used and unknown even to native speakers. In the process of learning we could not assume that students know even very frequently used phrases or how to use them correctly. In the matter of phrases the translation may not be the right expression. Translators do not translate a phrase but search for an equivalent phrase in the other language. But we use the term phrase translation since in general public it is more accustomed.

Most of contemporary users of computers use word processing software with spell assistant and grammar correction. Interesting enough is the fact that the product from the same manufacturer (Microsoft) works differently in Windows and MacOS. Is it just data gathering about the users of different operating system or is it something deeper? In the Windows world, "which" is a preferred word, but in the MacOS the preferred word is "that" (see Figure 1).

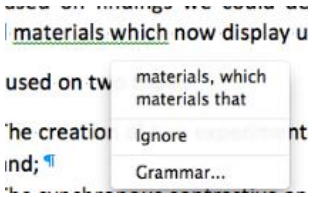

Figure 1: Mac OS Word 2011 give suggestion to use "that" but Windows Word does not show any grammar considerations in the same sentence.

2 Project description – phraseology of German language [3]

In 2011 the government funded German-Slovene contrastive research project started. The main focus of the research is the phraseology of the German language, which is synchronous with the contrastive and intercultural aspects compared with the phraseology contact in Slovene language. The project is based on the fact that phraseology of German and Slovenian language contrastive were only fragmentarily explored. Therefore we do not have theoretically and methodologically justified, empirically verified, and credible research results that would show a complex interlinguistic situation between German and Slovene language. The deficit in contrastive research is due to unreal binding of the two languages and historical influence on interlingual and intercultural aspects. History of Slovenian linguistic research is significantly related to German-speaking countries linguistics at least until the second half of the 20th century. Projects' research topics are therefore important for scientific linguistics in the Slovenian area and expected results are going to be a phraseological contrastive analysis of the two languages. Results will enrich the knowledge of the Slovenian language on comparative aspects. On the other hand results will be also useful in applied linguistics. Based on findings we could design better bilingual Slovenian-German dictionaries and other helpful materials which now display unwanted deficiencies.

Project is focused on two topics:

- The creation of two experimental corpuses German-Slovenian phraseological material and;
- The synchronous contrastive analysis of the material collected through formal syntactic, semantic, pragmatic, cultural, sociolinguistic, lexical and cognitive-psychological aspects will be applied to current theoretical frameworks and established research methods [4].

Expected results show a high degree of originality, a novelty in the current Slovenian linguistics and thus significantly reduce the gap in synchronous contrastive research on this language pair:

- Methodology of corpus acquisition of phraseological language data for the two languages,
- Qualitative and quantitative upgrade of existing collections of German and Slovenian language phraseological material.
- Contrastive typology of phraseological variation proverbial constructs.
- Contrastive typology of syntactic connectors for integration proverbial structures in the textual environment.
- Contrastive semantic analysis of collected cultural materials.
- Contrastive analysis of collected material from aspects of sociality and genre.
- Clarification of selected phraseology integration processes problems in the language acquisition and / or language learning,
- Clarification of selected problems lexicographic treatment of phraseology.

Results of research are expected to be the following:

- They are going to be important as the fundaments and pathfinder for further research on both languages contrastive.
- They are going to be useful in lexicographic practice, particularly the improvement of the quality of bilingual or general German-Slovenian dictionaries and language databases.
- They are going to be important for the learning and teaching of both languages as foreign languages in terms of systematic and rational integration of phraseological (and also cultural) content in language learning and teaching.

Application currently contains more than 2000 phrases (42% Slovene and 58% German). Project is going to be maintained and updated by students and professors from the Faculty of Arts at University of Maribor to provide a long time service.

This project has the similarities with two previous projects that serve as knowledge and structure background. Those projects were SprichWort [5] and EPHRAS [6]. In many interviews and internal education we learned enough to prepare software support for the project. To get familiar with the topic we had to study articles about electronic dictionaries [7]; language data presentation [8]; cross-language information retrieval [9] [10]; and machine translation [11] [12] [13].

Software module that works in web environment as standard web application or as web service was designed and implemented. ERM of the structure that enables linguistic analysis and different languages was prepared and implemented in MySQL (see Figure 1). In the initial considerations we have doubts should use transactional database. But later reviews show that MySQL is more than sufficient in this occasion.

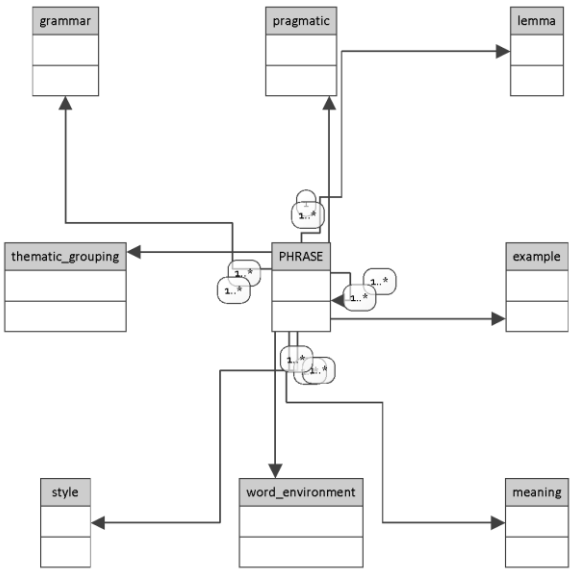

Figure 2: ER diagram for our project.

3 Phrases vs. words vs. sentence

Despite the fact that phrase is a concatenation of words it is generally not a sentence. Watching the phrase from its words perspective more than just occasionally it has ambiguous meaning. This fact has negative implication that the same phrase can in general have a different meaning if used in different sentences (word environment, pragmatic). Automatic translators generally have problems even with the well-known phrases (see Figure 3).

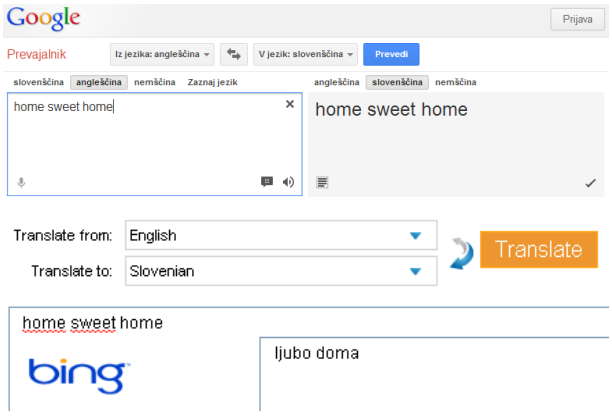

Figure 3: Well-known English phrase "home sweet home" Google translates to Slovene as original text (inadequate) but Bing translator is correct ("ljubo doma").

Explanation of the phrase depends on the context where the phrase is used. The system can run even without explanation but this would mean that phrase is not jet validated. Additional explanations can be added to the phrase as needed.

Phrase can have multiple synonyms in the same language but also in different language. We decided to address synonyms in the same language as variants. It was proven more adequate and precise in the communication. One phrase can have many synonyms in another language depending on the context where the phrase is used. From the literature it is evident that we should pay attention in this semantic structure [14]. In the beginning we were misled thinking that even terminology is part of the phrases [15]. We were wrong but the results enable us to better extract data from the existing projects.

Phrase is also linked to the context. Context of using the phrase is defined in the explanation (meaning, word environment, style, pragmatics, and examples). For more unambiguous search results we also use lemmatization which present additional set of problems.

4 Search topics

Since search is fundamental in our program we have decided to follow established rules. Our data are specific and we do not need to implement full search as on web search engines [16] but only those that suits our needs.

Searching words is implemented in a priority list. First displayed results are "**exact match**", then "**all words**" and at the third level "**first word first**" and so on. Since we do exact search anyway we do not need to implement exact search in quotes. Despite the fact that we have not thought about the exclusion the demand was addressed from the project members and therefore an exclusion of words can be entered to the search field using the minus character in front of the word.

Implementation of Boolean operations in the search was not considered relevant at the beginning. Later the possibility was analyzed (interviews with students on computer lab work) but dismissed due to users' preferences where most of users do not want to use Boolean operators anyway.

4.1 Preparation of data for search

In the general we could say that our problem is just another full text search problem which was already solved with many algorithms [17] (naïve string search, finite state automation based search, stubs, indexing and fuzzy approach). But in our case we have strings in the database. We therefore only search substrings inside single string at the time and since the number of records is small we were more than satisfied with the simple database build-in search.

The primary entity of the search is phrase. In the case when user enters exact phrase into the search engine it would be very simple to find proper results. But the nature of searching is different and students learning the topics in most cases do not know the phrase exactly. The translator has little less problems with the phrase since he sees the phrase's text in the original language and tries to find proper translation.

In the database we have multiple attributes about the phrases: *grammar*, *pragmatic*, *thematic_grouping*, *example*, *style*, *meaning*, and *word_environment*. All these attributes are necessary for linguistic analysis but not so much for searching purpose. Later we discovered that with this additional data a metrics for evaluation of the search results could be refined. From the perspective of the user friendliness it would be unwise to make user interface with multiple inputs for different keywords in each of the field for search result. All project partners

have dismissed such approach as "not user-friendly" behavior.

4.2 Problem of lemmatization

In some cases search of a phrase is not efficient since the words do not depict its primary meaning. The lemmas instead of words from the phrase would be much better in these occasions. Despite the fact that software lemmatizers are available (even as software service [18]) automatic lemmatization of the phrase is not an easy task. We have found many faults in the automatic lemmatization of phrases therefore user (author of the phrase) is required to authorize the lemmatized phrase. This is actually simple task for a human. In our case we use automatic lemmatizer (LemmaGen) that prepares suggestion and author verifies the correctness of the results. In reality most of the time results of lemmatization are correct, but occasionally some words need to be changed by authors to achieve proper lemmatization of the phrase.

4.3 Searching equivalent phrase

Search is implemented on entity phrase and lemma (see Figure 2). After software testing it was discovered that language specific search implementation is not need. If the word is in one language we try to auto detect language just for the display of results. In general we perform search through all phrases in the database and received results are displayed based on priority list (metrics). In some cases the problem arose when the word is the same in the different language with different meaning. It would be vise not to search on conjunction words but for the statistical purposes and text analysis this is also required. Sometimes funny situations occur in the search. Word "in" in Slovenian is actually "and" in English; but "in" in English is "inside" translating back to Slovenian. Therefore we need to search all entered text and metrics should be smart enough to guess the search language. If user is authenticated (logged into the system) the language is statistically defined from its previous searches. But we cannot make any relevant prediction if user is anonymous and search text consist of only one word that is the same in many languages. From the results of test users (students of language study programs) problem is not that obvious though.

For a translator and student it is quite easy to translate the phrase if he/she has access to our software. It just finds the phrase in the original language (either by retyping or copying text) and then it picks the function for synonyms and "voila". If in doubts he can always check the context and examples.

4.4 Problems of typing errors

It is not wise to assume that users will enter their search text in absolutely correct style. Typing errors are common in the text in languages where the spelling is not equal to the pronunciation. Many word processors have built in check spellers. It is nice addition to the writer and most people who have to write in the language that is not theirs mothers tongue actually depend on them. But it is not limited only to these occasions typing errors are frequent and generally not detected by the author of the text. Check speller is just a tool and is not successful in the recognition of the wrong word that is spelled right. In the sentence: "*I see that you **two** made the same mistake.*" "*two*" could be replaced with "*too*" and would be still undetected not just by check speller but also from the reviewer in context is not recognized correctly. Many word processors have even grammar assistant mode. In general this is a good idea and is helpful but sometimes it is just wrong and many English linguists are not satisfied with them at all (discussion at the conference of ED-MEDIA 1999).

If users satisfaction should be enhanced than a use of statistics is necessary. The software learns from users' interaction and in time a large amount of general typing errors can be detected. In our perspective the search statistics have to be implemented for educational purpose but we used it for the better recognition of user's intention when we assume the typing errors in the input field.

Typing errors are only one type of problem of misspelling. We have discovered by observing our students that users sometimes cannot spell the correct word but they just enter the word as it should sound. In such cases a vocal synonyms could become really handy. In our occasion it was a bit beyond the scope of our project but research shows that Google made even vocal predictions in their search engine (see Figure 4).

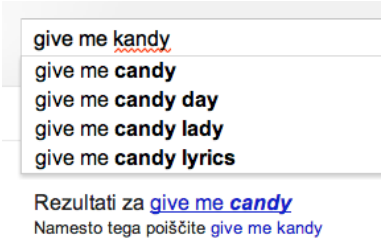

Figure 4: Vocal prediction in Google

4.5 Problem of ranking the search results

Previously we have mentioned the problem of the matrices for search results. Searching the results we may get a list of answers. For user friendly results it would be wise to ordered them according to the metrics that satisfy users' need. Therefore "How to implement a proper metrics?" is a valid question. In our case we decide to do the following:

One word question: In one word question all search results are equal and there are no distinction which answer is more appropriate than other. But since we have statistics of the search we can assume that more observed answers are more likely to be the right one.

Multiple words question: In the two words question we have multiple possibilities to present answer. Most obvious is the exact match which is the first order inside more answers we apply statistics. Second order is the all words search but in this occasion it is

possible that searched words can be more or less apart. Therefore it may be second level of metrics where the average length between words is calculated and the result with the lowest average length is displayed first. On the other hand we can also get results where not all words are present in this situation more words that match the search string is displayed first.

The search is therefore not entirely trivial even in these two examples. To complicate matter even more we have to know that it is possible that in the search field user enters ***delimiters***. How to address the problem of delimiters (like ***commas*** and ***dashes***) depends on the occasion. Delimiters can be entered into the search field by mistake or intentionally therefore we cannot dismiss them altogether. Commas are little lesser problem but a dash actually changes the meaning of the adjacent words. Since we cannot know the intention of the users it would be wise to search for the string as entered and later by extracted words.

5 Fuzzy user friendly implementation

From the implementation viewpoint we have to know that in the database we have phrases and not individual words. Searching therefore is not just a simple SQL query but needs to be additionally processed to implement multiple queries and consolidate results. Ruby on rails facilitates this with *ActiveRecord* which runs queries only at the time of presenting data. The logic can be more complex but still very efficient.

To make searching more accurate we are mixing a regular database search with gathered statistics. For every search query, we are recording the search phrase and the following action from the user. With this simple data pair we can gather what the user was searching for and what he considered the correct result. This creates a ranking mechanism which the users can turn on or off depending on their preference. This way we can recommend frequently selected phrases based on the search phrase.

Search in advance and search assistant. *ActiveRecord* is a tool that enables searching in the background. The same search mechanism is used in the Google web search. As user type into the search box a possible results are displayed. This approach is feasible but was dismissed since one word search gives too much results and just distract our intended users.

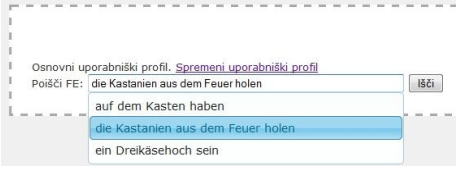

Figure 5: Active Record search.

6 Results

After authentication our web software is ready for search in the Figure (see Figure 6). Search field accept string with the addition of special characters.

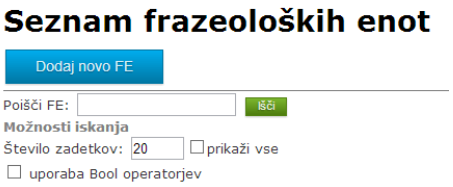

Figure 6: Search field for the web software

If only one word is entered in the search field then the search result display all occurrences of this word in the database (see Figure 7).

Figure 7: One word search.

In case when two or more word are entered in the search field the results shows: (1) exact word matching; (2) intersect between these words; and then (3) union of these words.

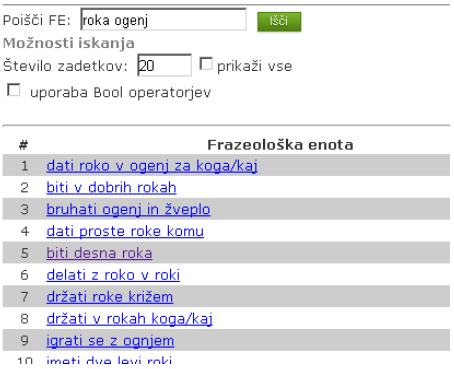

Figure 8: Two word search.

If Boolean operators are enabled then the special notation is used. A plus sign before the word means that word need to be in the search result (like intersect) and only results which satisfy the question are displayed (see Figure 9). In some occasions it may happen that the search results do not correspond to the natural thought of the user. Users have a tendency to enter the phrase as a sentence. They often think that first word is also fixed and the rest are added to refine the search of the first word (see Figure 10). In general this is not the true if we use simple database text search. Results show that no matter of the position of the word in the search field we got only one fixed word and another as open. It is

necessary to change the search string to fix two word (see Figure 11) to get the result we want.

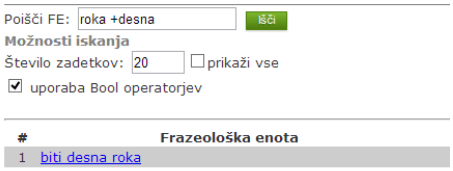

Figure 9: Boolean search

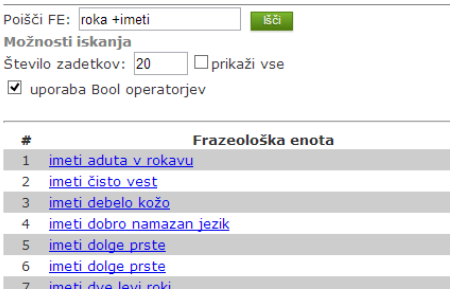

Figure 10: One open and one fixed word search.

Figure 11: Boolean search with two fixed words

In case that we want some word not to be present in the search result a minus should be placed in front of this word (see Figure 12).

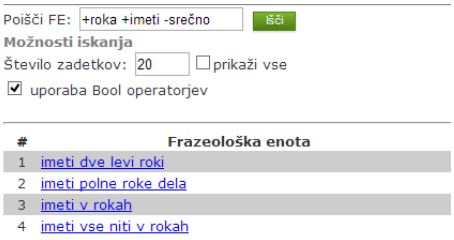

Figure 12: Intersect and exclusion search

For the purpose of our software this search engine is sufficient but if we are going to expand the product to satisfy not just phrases than we will need to make additional adjustments.

7 Conclusion

In any text related databases a search mechanism is inherently complex. Search as we know it from the programs (like Word, Excel...) are not suitable since they all search substring in the set of strings. More appropriate examples for the text search are web search servers (like Google, Bing…) and they can be blueprint for successful text search engine. In the linguistic projects a subset of all available search categories is needed. Advanced search is rarely used even in the web search and the same is also true in linguistic projects. Students generally have no need to have Boolean search implemented. But after a while almost anyone find the need for exclusion operators. The search engine should have also language specific part or module. This module would based on statistics and find the right results even if user enters wrong spelled keywords or if it is entered vocally suitable keyword.

In the project (*Phraseology of German language*) we have implemented only those search categories that suit our need. Statistics is important for registered and anonymous users. Registered users actually with their searches and selection of results teaches the system about their preferences on the other hand anonymous users are all treated alike and only common statistics can be used. It is possible to implement even typing error correction and vocal search in our project based on statistics and prediction module but till now we have no use of them. The potential problem of separators was not detected in the phrases but in other linguistic project it would definitely become evident.

8 Literature

[1] EU legislation, "Europa - Summaries of EU legislation," 3 3 2011. [Online]. Available: http://europa.eu/legislation_summaries/education_training_youth/lifelong_learning/c11090_en.htm.

[2] E. R. Westfall, "Machine translation and the information soup," *Lecture notes in artificial intelligence,* vol. 1529, pp. 501-505, 1998.

[3] Faculy of Arts, "Frazeologija nemškega jezika," University of Maribor, Faculy of Arts, 2011. [Online]. Available: http://projects.ff.uni-mb.si/frazeologija/. [Accessed 14 4 2013].

[4] P. Ďurčo and V. Jesenšek, "Sprichwörter mehrsprachig und korpusbasiert in einem multilateralen EU-Projekt," *Slowakische Zeitschrift für Germanistik,* vol. 1, no. 1, pp. 63-73, 2009.

[5] V. Jesenšek, M. Fabčič, K. Steyer, K. Hein, P. Ďurčo, D. Chovaniková, T. Forgács, T. Kispál, L. Marek and V. Kozáková, 2008-2010. [Online]. Available: http://www.sprichwort-plattform.org/.

[6] V. Jesenšek, R. Muhr, D. Helic, A. Borgulya and P. Durco, 2004-2006. [Online]. Available: http://www.ephras.org.

[7] H. Bergenholtz, T. Bothma and R. Gouws, "A model for integrated dictionaries of fixed expressions," in *eLex*, Bled, Slovenia, 2011.

[8] H. Begenholtz, Needs-Adapted Data Access and Data Presentation, P. A. Fuertes-Olivera and H. Bergenholtz, Eds., London: Continuum Intl Pub Group, 2011.

[9] J. Wang and D. W. Oard, "Matching meaning for cross-language information retrieval," *Information*

processing & management, vol. 48, no. 4, pp. 631-653, July 2012.

[10] R. Mihalcea and D. Radev, Graph-Based Natural Language Processing and Information Retrival, NY: CAMBRIDGE UNIVERISTY PRESS, 2011.

[11] J. Coleman, Introducing Speech and Language Processing, Cambridge: CAMBRIDGE UNIVERSITY PRESS, 2005.

[12] M. Farrus, M. Costa-jussa and M. Popovic, "Study and Correlation Analysis of Linguistic, Perceptual, and Automatic Machine Translation Evaluations," *Journal of the american society for information science and technology*, vol. 63, no. 1, pp. 174-184, January 2012.

[13] P. Koehn, Statistical Machine Translation, CAMBRIDGE UNIVERSITY PRESS, 2010.

[14] M. Smistson, D. V. Busescu, S. B. Broomell and H.-H. Por, "Never say "not": Impact of negative wording in probability phrases on imprecise probability judgments," *International journal of approximate reasoning*, vol. 53, no. 8, pp. 1262-1270, November 2012.

[15] C. Jacquemin, Spotting and Discovering Terms through Natual Language Processing, Massachusetts Institute of Technology, 2001.

[16] Google, [Online]. Available: http://support.google.com/websearch/bin/answer.py?hl=en&hlrm=sl&p=adv_operators&answer=136861.

[17] L. Benuskova, "COSC 348: Computing for Bioinformatics," n.d.. [Online]. Available: http://www.cs.otago.ac.nz/cosc348/alignments/Lecture04_StringSearch.pdf. [Accessed 15 4 2013].

[18] Josef Stefan Institute, Ljubljana, Slovenia, [Online]. Available: http://lemmatise.ijs.si/Services.