the ESPecialist, vol. 31, nº 1 (49-67) 2010 ISSN 0102-7077

A PROGRAM FOR FINDING METAPHOR CANDIDATES IN CORPORA
Um Programa para Identificar Candidatos a Metáfora em Corpora

Tony Berber Sardinha* (Pontifical Catholic University of São Paulo – PUC-SP, São Paulo, Brazil)

Abstract

In this paper, I present a computer program ('Identificador de Metáforas') for finding metaphor candidates (i.e., words that are likely to have been used metaphorically) in corpora. It works by matching each word in the corpus to five databases that contain several kinds of information about lexis and its relationship to metaphor, all extracted from extensive hand-annotated corpora. These databases store the probability of words being metaphorical based on their previous use in metaphors, on lexical patterns occurring near and around words, and on their word class. This article explains how the program was created and how it works.

Keywords: *conceptual metaphor; linguistic metaphor; Corpus Linguistics; metaphor identification.*

Resumo

Neste trabalho, apresento um programa de computador ('Identificador de Metáforas') destinado a encontrar candidatos a metáfora (i.e., palavras que possivelmente foram usadas metaforicamente) em corpora. Ele funciona comparando cada palavra do corpus a cinco bases de dados que contêm vários tipos de informação sobre o léxico, colocações e classe gramatical, retiradas de corpora anotados manualmente. Essas bases de dados registram a probabilidade de uma palavra ser usada metaforicamente a partir dos padrões de que ela faz parte. Este trabalho explica como o programa foi criado e como funciona.

* The author wishes to thank CNPq (Brasília, Brazil) for grants # 350455/03-1 and # 307307/2006-9, and CAPES (Brasília, Brazil) for grant # 0397/04-0. The author is personally indebted to Dr Doug Biber for his help on this project during a six-month visit to Northern Arizona University.

Palavras-chave: *metáfora conceitual; metáfora linguística; Linguística de Corpus; identificação de metáfora.*

1. Introduction

With the publication of 'Metaphors we live by' by Lakoff and Johnson (1980), metaphors have been shown to be ordinary devices used in everyday communication, instead of markers of sophisticated or literary style only. Metaphors are part and parcel of the way regular people think. Hence, according to Lakoff and Johnson, people normally live by such (conceptual) metaphors as LOVE IS A JOURNEY or ARGUMENT IS WAR, which are revealed by the regular use of expressions such as 'We hit a dead-end street' or 'He attacked every weak point in my argument', respectively.

It is only recently that metaphor analysts have begun to turn their attention to the analysis of metaphors in corpora (e.g.Deignan, 1999). And as they do so, they encounter problems that were unforeseen by the analyses carried out by Lakoff and Johnson, who based their arguments on few selected examples. If language is full of metaphors, then so must be corpora, as they are language samples. That metaphors are out there in corpora, everyone knows, but how to get to them is a different matter.

Back in 1991, Martin (1991) observed that 'unfortunately, there are no robust automated metaphor tools analogous to part-of-speech taggers … that would reliably permit large-scale automatic analysis' (: 16).Today, the situation has not changed much with respect to the availability of such a tool. The resources available for corpus-based metaphor analysts to comb through a corpus for metaphor candidates are restricted to those generic tools available for traditional corpus linguist research, such as concordancers, wordlisters, chunkers, frequency markedness identifiers, and part of speech taggers, even though the needs of metaphor analysts are quite specific. For one thing, metaphor analysts need to know if a word or expression is a metaphor, a category that is so fuzzy that the odds of two people agreeing on the metaphoricity

of a word is quite low (Cameron, 2002). For another, we know as yet very little about several essential properties of metaphors, such as what they look like, what their length is, what words seem to signal them, what their distribution is like in a text, what the odds are of finding them in any given corpus, and so on, for a researcher to be able to enter a 'metaphor word' with confidence into a concordancer or tell which words are metaphoric by looking at a frequency wordlist or keyword (marked frequency words) list.

Hence, because of these problems, it is not surprising that many metaphor analysts prefer to read parts of the corpus beforehand, identifying a few metaphors and the words associated to them, and only then carry out searches for these words in the whole corpus with a concordancer. This strategy has been used and proved helpful in previous research (Charteris-Black, 2004), but there is no guarantee that any small portion of a corpus is likely to give us a large number of candidate metaphors, or that the researcher has not missed any metaphors during his reading. If they decide to read larger portions of their corpus, then another problem presents itself, namely that it gets harder to apply metaphor identification criteria consistently as the amount of data grows. It would seem, then, that metaphor identification in corpora is a task that computers might be well suited for, as they neither suffer from fatigue nor do they shift identification criteria along the way.

A computer program for processing an entire corpus without an initial set of metaphorical words or metaphorical expressions and that could yield a list of potential metaphors would thus seem to be a welcome addition to the set of tools currently available to metaphors analysts. This would free the analyst from the tedious job of reading a whole corpus and marking up metaphors, or from running huge numbers of concordances only to find out that a minority of them actually returns metaphors.

This paper describes the development of one such tool (the 'Identificador de Metáforas', or Metaphor Identifier), which is intended as a device for signaling *candidate* metaphors, that is, words that were probably used metaphorically in corpora. It works by matching each

word in the corpus to five databases (described below) that contain several kinds of information about lexis and its relationship to metaphor, all extracted from extensive hand-annotated corpora. It is important to stress that the program offers candidates for metaphor only, and that it is only the analyst that can ascertain if a candidate is indeed a metaphor. It is hard to imagine the process of identifying metaphor being fully automated, since metaphor identification is highly context-dependent, and it is only by taking into consideration the context in which a word or expression was used that one can state with confidence if any stretch of language is metaphorical or not. Computers cannot deal with the context of language use, hence they would not be able to replace the human analyst in making judgments about the metaphorical meanings of words and expressions.

The program here assigns a probability of metaphoricity (the likelihood of a word being part of a linguistic, i.e. not conceptual, metaphor) to each word in the input file. The process of gathering information to arrive at such probabilities is described further below.

The program was designed to handle corpora in Portuguese only, at this stage. This is because the corpora that were used to train the program were in Portuguese as well. The program itself, however, can handle texts in any Western language, so long as the necessary training resources are available.

The program is part of the corpus analysis tools at CEPRIL (www2.lael.pucsp.br/corpora), the Center for Research, Resources and Information on Language of the Graduate Program in Applied Linguistics, Pontifical Catholic University of Sao Paulo (www2.lael.pucsp.br).

2. Previous computational approaches to finding metaphor in corpora

In the literature, there are at least two other computational tools for identifying metaphors in corpora. These are described below in chronological order.

Berber Sardinha (2002) describes a collocation-based method for spotting metaphors in corpora. His procedure is based on the notion that two words sharing collocations in a corpus may have been used metaphorically. For example, in a corpus of Portuguese academic writing, 'ensino' (teaching) and 'construção' (construction) shared a number of identical collocates, which was later interpreted as a conceptual metaphor ('TEACHING IS BUILDING'):

processo (process)
visão (vision, view)
relação (relation, relationship)
linguagem (language)
escrita (writing)
processos (processes)
conceitos (concepts)

As can be expected, a potentially huge number of words share collocates in a corpus. Therefore, the first step was to pick out a reasonable number of words that had an initial likelihood of being part of metaphorical expressions. First, words with marked frequency (in relation to a large general corpus of Portuguese) were selected. Then, their collocations were scored for closeness in meaning using a program called 'distance' (Pedersen & Patwardhan, 2002), under the assumption that words involved in metaphorical expressions tend to be denotationally unrelated ('time' and 'money', 'love' and 'journey', etc). This program accesses WordNet in order to set the scores for each word pair. The scores had to be adapted in order for them to be useful for metaphor analysis. Finally, those words that had an acceptable semantic distance score were evaluated for their metaphoric potential. The results indicated that the procedure did pick up some major metaphors in the corpus, but it also captured metonyms.

Another approach to finding metaphor in corpora is CorMet, presented by Mason (2004). It works by searching corpora of different domains for verbs that are used in similar patterns. When the system spots different verbs with similar selectional preferences (i.e., with similar words in subject, object and complement positions), it considers them

potential metaphors. For instance, if a corpus of the finance domain has verbs such as spend, invest and flow that collocate with words related to money (capital, cash, money, etc) and a corpus of the chemistry domain has the same verbs collocating with liquid-related words (substance, fluid, liquid), then this means that this may be a metaphorical mapping such as MONEY IS LIQUID, which in turn gives rise to sentences such as 'Capital flowed in the new company'.

CorMet requires specific domain corpora and a list of verbs for each domain. The specific domain corpora are compiled by searching the web for domain-specific words. These words are selected by the author, based on his previous knowledge of subject areas and are stemmed. The most typical verbs for each specific corpus are identified through frequency markedness, by comparing the frequencies of word stems in the domain corpus with those of the BNC. The resulting words have a frequency that is statistically higher in the domain corpus than in the reference corpus. These stems are then classified according to part of speech by consulting WordNet. CorMet's performance was tested against its ability to detect mappings in the Master Metaphor List (Lakoff *et al.*, 1991). This was done subjectively, by having the author match the examples pulled out by CorMet with those on the metaphor list. The success rate was 77%, with CorMet being able to identify 10 out of 13 major metaphors.

3. Overview of the program

This program has several components, which are summarized below. Each of these elements is described in more detail in the section dealing with the development of the program.

The components of the program are:

1. A target corpus: the file (one or more texts) that the user wants to tag. This is an ASCII file that is processed by the program and is output in several formats:
   a. Word list: a list of word types in the corpus, untagged and unlemmatized.

b. Left-side bundles: a list of three-word sequences of contiguous words occurring immediately to the left of each word form in the input corpus. Example: 'a evolução do [word]' ('the evolution of …').
c. Right-side bundles: a list of three-word sequences of contiguous words occurring immediately to the right of each word form in the input corpus. Example: '[word] a chance de' ('… the chance of')
d. Frames: a gapped three-word list, in which there is a word followed by a gap followed by another word. Example: 'A [word] do' ('the … of').
e. Word classes: a list of word types in the corpus and their part-of-speech tags, drawn from a POS-tagged version of the corpus.

2. A training corpus: a corpus used to extract information about the probability of metaphor vehicles occurring in specific patterns and as specific word classes. A metaphor vehicle is a word that is the focus of a metaphorical expression. This information is organized as information databases.
3. Knowledge bases: Sets of data accessed by the program, from which it draws the probability of word being a metaphor vehicle.
4. Tagged corpus: A version of the target corpus tagged for metaphors. This is a list of all word forms found in the target corpus, each followed by a tag. The tag is a number that indicates the probability of that word being a metaphor vehicle. The probability of a vehicle is equal to the number of times that word was coded as a metaphor in the training corpus divided by its overall frequency in the same corpus. The program's output is illustrated below. The output shows that for that particular corpus, the word with the highest probability of being part of a metaphor is 'aumentar' [increase], with a 34.16% chance of being metaphorical, followed by 'comunidades' [communities] and 'exigência' [demand], both with 33.69%, 'mercado' [market] with 33.61%, and so on, down the list.

TEXT TAGGED FOR METAPHOR
Sorted by probability
Sun Jun 19 17:41:53 BRT 2005

#	Word	Tag
000001	aumentar	.341620000000
000002	comunidades	.336900000000
000003	exigência	.336900000000
000004	mercado	.336120000000
000005	passou	.332240000000
000006	analfabetismo	.314660000000
000007	parceria	.309400000000
000008	passo	.306120000000
000009	adoção	.287620000000
000010	acesso	.276900000000

The main steps taken in the development of this program are these:

1. Setting criteria for the identification of metaphors in corpora. This was based on previous literature and followed broadly the applied linguistic approach developed by Cameron (2002).
2. Identifying metaphors in a small register-specific corpus by hand: each word was manually coded as metaphorical or as non-metaphorical. More specifically, words were coded as a metaphorical if they were considered a metaphor vehicle in a linguistic expression. Upon completion of this phase, 4,385 lines had been coded and a pool of vehicles emerged, containing 423 words.
3. Extracting concordances from a much larger corpus for each word in the pool and subsequent hand coding of these lines.

This was needed in order to strengthen the probability profile of each word. These concordances were later combined with the set of concordances from the previous corpus, producing a definitive set of 21,928 coded concordance lines.

4. Developing databases using information from the definitive set of concordances. These are the modules from which the program draws information in order to operate, and each holds specific information about the probability of metaphors.
5. Designing, programming, running and debugging the program.
6. Hand coding texts to be used for evaluating the program performance.
7. Proposing ways of improving the program in the future.

The remainder of the paper is organized around these topics, in that order.

4. Criteria for the identification of linguistic metaphor

The manual analysis was based on the notions of both linguistic metaphor (Cameron, 1999; Steen, 1999) and conceptual metaphor (Lakoff & Johnson, 1980). In coding the corpora, an expression was considered a linguistic metaphor if it was possible to detect a conceptual metaphor underlying it. A conceptual metaphor is a mapping between two conceptual domains (Lakoff & Johnson, 1980), the source and the target domains. For instance, the linguistic metaphor 'he wastes a lot of time' is a realization of the conceptual metaphor Time is Money; time is the target domain, and money is the source domain. This conceptual metaphor holds true in Western Culture, so that time is conceptualized as money: it can be wasted, lost, saved, invested, managed, and so on. As a result, it is unwise to waste time as it is to waste money, even though people do not actually waste anything concrete if they waste time, since time cannot be held, stored or put in one's pocket.

In the analysis of linguistic metaphors, two components were distinguished: topic and vehicle. The topic is that part of the linguistic

metaphor that represents what the metaphor refers to. In an expression such as 'he wastes a lot of time', 'time' is the topic. The vehicle, by contrast, is the metaphorical focus (Cameron, 2002, p.10), which means that this is the component that is used metaphorically. In the previous example, 'wastes' is the vehicle. The coding was restricted to the vehicles only, as these are both the focus and the obligatory element in a linguistic metaphor.

The identification of metaphors was carried out according to the applied linguistic approach developed by Cameron (2002). She summarizes this approach as follows:

> 'The category of linguistic metaphor will be established through the potential for incongruity between two domains to be interpreted from surface lexical content. Neither metaphorical intention nor metaphorical interpretation will be necessary conditions for membership.' (: 25)

To illustrate, let's see the example below, from the training corpora:

> 'em função da volatilidade que o mercado tem apresentado nos últimos meses' (given the volatility that the market has shown in the previous months)

The word 'volatilidade' (volatility) is the vehicle in this passage. It was considered a vehicle because it stems from a conceptual metaphor, namely 'Behavior Of The Economy Is The Behavior Of Gas' (Charteris-Black, 2004), which signals a mapping between properties of gas and properties of the economy. There is incongruity (Cameron, 2002) or tension (Charteris-Black, 2004) embedded in it, since chemistry and economy are two separate domains that are brought closer together by the metaphor: markets (the target domain) are conceptualized in terms of the domain of chemistry (the source domain). In chemistry, volatility is a measure of 'how quickly a substance forms a vapor'[1] but for the market,

1. http://www.cdc.gov/od/ohs/manual/chemical/chmsaf5.htm

volatility means the fluctuation of an index, how rapidly it goes up and down. There is incongruity here since turning into vapor is not quite the same as fluctuation (which is in itself another metaphor!). What these two meanings have in common is rapid change, which happens to both liquids turning into gas and to market indexes going up and down. What is interesting is that traders, brokers and so on do not need to go back to the source meaning of volatility to make sense of it in their domain. That is, they do not need to process 'volatility' as a metaphor when they use it in order to understand it.

This is important to bear in mind when coding metaphors in corpora: the analyst does not need to infer whether an expression was processed as a metaphor in the context by those who heard or read it, nor whether it was deliberately intended as a metaphor by those who produced it (Cameron, 2002: 12). In the previous example, it is most certainly the case that users in the domain of market did not think of 'volatility' as a metaphor, that is, they did not activate the source domain of chemistry in order to make sense of the meaning of the word. Linguistic metaphors that do 'activate domains in the mind of a discourse participant, and that lead to the noticing of incongruity' are called process metaphors (Cameron, 2002: 12). In the applied linguistic followed here, this did not cause a problem, and volatility was marked as a metaphor vehicle, on the grounds that it was possible to envisage a mapping between two distinct domains.

5. Preparation of the program

The basis for the program are two corpora that were hand-coded for metaphors following the principles described above.

The first is a corpus of conference calls held in Portuguese by an investment bank in Brazil. This corpus has 85,438 tokens and 5,194 types. In the conference calls, investors, bank executives and the press talked about matters related to stocks, budgets and investment trends. These calls were recorded and transcribed by the bank and were made available on the web. I downloaded them and then coded them for metaphors three times. These subsequent coding sessions were carried

out as a means to ensure the reliability of the manual annotation. There are two main reasons why I chose this corpus. Firstly, the program is part of a larger ongoing business language project[2]. Hence, it was motivated by a need to locate metaphors in the business domain. Secondly, having a small constrained business corpus would probably help during the manual identification process, as the number of individual coding decisions might be fewer compared to those needed to code a register-diversified corpus, since metaphorical expressions would tend to be less varied in the register-specific corpus. At the end of this analysis, the total number of word forms being used metaphorically was calculated, and for each word, their frequency as metaphor a vehicle and as a non-vehicle was also computed. There were 441 word forms that were used at least once as a metaphor vehicle, totaling 4385 occurrences. A problem with these is that 157 (35.6%) appeared only once. This large number of low frequency words would generate a large number of unreliable probabilities, as they would be based on single cases. To remedy this situation, another training corpus had to be utilized, this time larger than the initial corpus, so that it would yield more cases of each vehicle.

The second training corpus was the Banco de Português (Bank of Portuguese; http://lael.pucsp.br/corpora/bp, a large, register-diversified corpus, containing nearly 240 million words of written and spoken Brazilian Portuguese, compiled by members of the DIRECT Research Group (www2.lael.pucsp.br/direct). From this corpus, concordances were made for each of the 441 vehicles found in the analysis of the previous corpus. A total of 17453 concordance lines were extracted and coded. In the process, the metaphoricity of 19 of the previous 441 vehicles was reconsidered and these vehicles were discarded, leaving a final count of 422 vehicles.

The frequencies taken from the Bank of Portuguese and from the conference call corpus were then joined, so that the frequencies for each vehicle reflected their use as metaphors or non-metaphors both in the specialized business corpus and in the general corpus. None of the vehicles had a frequency of 1; the lowest frequency was 2, and the average frequency was 322.

2. For more information, visit http://lael.pucsp.br/corpora/bp

6. Developing the program

The program was written by the author as a Unix Shell Script, using mostly text utilities such as grep, sed, tr, and join, available in the Macintosh 10.3 Operating System. The program also contains code written in awk and Perl.

As far as speed, the tagger achieved 421.8 words per second on the server, on a 1,410,495-word corpus, which is its best figure so far. It must be said, though, that speed is not the main consideration at this stage in the development of the program.

Figure 1 below shows a screenshot of the program.

Figure 1: Home page of the program at the CEPRIL website (www2.lael.pucsp.br/corpora)

The knowledge databases that the program draws on for its analyses are plain text files, holding one record per line (corresponding to a word), with each line having two fields (a word or a group of words, as described further below). These databases were designed specifically for this program; none was an independent database that existed prior to the program. The hand-coded corpora were the raw material from which the information for the databases was obtained.

The databases are used as follows. For each word form in the target corpus (the file(s) submitted to it), the program tries to match it to an entry in the database. If there is a match, the value associated with that entry in the database is assigned to the word form. If there is no match, a value close to 0 (namely .0001) is assigned to that word form. After all words have been looked up in a particular database, the program moves on to the next database and starts the lookup process again, until all databases have been accessed. At the end of this phase, the program has a set of five probabilities for each word form in the target text. It then calculates an average probability for each word form and assigns that probability value (a figure from .0001 to 1) to the word form.

The first database is the 'vehicles database', which contains a field for each word and another field for its probability. This is the centerpiece of the program, and works as a lexicon in part of speech taggers. If the program finds the target word form in this database, it assigns its database probability to it. If it does not find the word in the database, it assigns .0001 as probability. Say, for instance, that the program comes across the word 'nível' ('level') in the target text. This word is the database, with a probability of 1, and so the program will assign probability 1 to 'nível'.

The second database is the 'left-side bundles database'. A bundle is a sequence of words as they are found in a corpus (Biber & Conrad, 1999); so a three-word bundle is a sequence of three words in the exact order in which they appeared in the source corpus. This database contains three-word bundles that appeared at least twice in the training corpora immediately to the *left* of a vehicle, as well as its probability of occurring in that particular slot. In order to access this database, the program first

extracts a list of three-word bundles from the target corpus, and then it looks for each input corpus bundle in the bundle database. If it finds it, it will assign the probability of that bundle to the word occurring next to it in the input corpus; if it does not, it will assign a probability of .0001 to it. For instance, assume that the three-word bundle 'a evolução da' (the evolution of) occurs immediately to the left of the vehicle 'inflação' (inflation) in the input corpus, and that 'a evolução da' appears in the left-bundles database with a probability of .9887. In this case, 'inflação' would be tagged as having probability .9887.

The third database is the 'right-side bundles database'. This is similar to the previous one, except that it contains bundles occurring at least twice immediately to the *right* of each word in the training corpora. The program first extracts a list of bundles occurring to the right of each word in the target corpus and then tries to match them with the database bundles. For example, suppose that 'a chance da' (the chance of) is included in this database with an associated probability of .8666, and that the expression 'a chance da inflação' (the chance of inflation) occurs in the input corpus. The program would assign the probability .8666 to 'inflação' (the word immediately to the right of the bundle). If the bundle being looked up is not found in the database, the program assigns a probability of .0001 to the target word.

The fourth database is the 'frames database', which contains frames occurring at least twice around vehicles in the training corpora. A frame is a gapped three-word sequence, such as 'the … of' (Renouf *et al.*, 1991). First, the program compiles a list of three-word frames for the target corpus. Then, it looks for each input corpus frame in this database and, if found, assigns the database probability to the word occurring in the center of the frame. If it does not find it, it assigns a .0001 probability. For example, if the expression 'o nível de' (the level of) occurs in the input corpus and 'o … de' (the … of) is included in the frames database with a probability of .6666, 'nível' (the center word) will receive the probability associated with that frame (.6666). If 'o… de' is not present in the database, the program will assign a probability of .0001 to 'nível'.

The last database is the word class database. This contains the probability of each word class being a metaphor vehicle in the training corpora. First, the program tags the target corpus for part-of-speech, using a version of QTAG (Mason, 1997) trained for Brazilian Portuguese. This program is available online at www2.lael.pucsp.br/ corpora. Then it looks up the word class for each target word in this database and assigns the probability of that word class to the target word. For instance, nouns have a probability of .6842 of being metaphor vehicles. A noun such as 'inflação' (inflation) would therefore receive a probability of .6842. Since all word classes are represented in the database, no target words fail to be matched to an existing probability in the database.

Table 1 below summarizes the information in each database.

Database	Entries	Average probability
Vehicles	423	0.6846
Left bundles	539	0.9772
Right bundles	602	0.9832
Frames	164	0.6289

Table 1: Entries and average probabilities in each database

As the table shows, the largest database is the right bundles database, with over 600 entries. This is also the one with the highest probabilities associated to the entries (about 98% on average).

At the end of this phase, each word in the input corpus will have received five scores. The next step is to produce a final score that will be the actual probability tag. This is done by averaging out the scores. The final tag is then the average probability of a word being a metaphor according to the five databases.

7. Conclusion

This paper presented a computer program for identifying metaphor candidates. The program is intended as a tool that can help

researchers find words that are more likely to be metaphor vehicles in a corpus. As such, it may be used as a device for signaling those words that the researcher might want to focus on first, because these have a higher probability of being metaphors in their corpus, or conversely, it may indicate those words that are worth looking at because of their apparent low probability of being metaphors. In any case, the program's output is arguably a better place to start than an ordinary wordlist of the corpus, or a list of words that the researcher may have an intuition about their metaphoricity in the corpus.

Tags are probability values assigned to word forms. Since the tags are not categorical (e.g. 'metaphor' or 'non-metaphor'), the program output cannot be used directly to pull out metaphors. A researcher would need to set his/her own cut-off point for metaphors. The program output is therefore not intended as a final analysis, but as a starting point for more detailed analysis.

The program is restricted to finding one component of linguistic metaphors (the vehicle, or metaphor focus), and does not identify or label conceptual metaphors (the abstract conceptual structure underlying linguistic metaphors). Because the program tags individual words, it does not indicate where metaphorical expressions begin and end. This feature might be incorporated in future versions, but this will depend on further coding and delimiting of metaphorical expressions in corpora.

The program is highly dependent on five specially-crafted knowledge bases, which come from hand-coded training corpora. At its current state, the program has been trained on business texts in Portuguese, and so it is restricted to that kind of text. The knowledge databases did seem to perform well, on average, with the share of each one in identifying the actual metaphors in the test corpus varying from 15% to 35%. The one database storing words that have been metaphors in previous text holds at the moment just over 400 words only. This is clearly a small lexicon, and so there is a need for guessing possible metaphors based on their lexical patterns and word class. This is the job of the other four databases. Together, these databases accounted for about 65% of the correct guesses made by the program.

The version of the program presented here is still under development. It will be updated continuously, as more data are hand-coded for metaphors. It is hoped this program will be of help to metaphor researchers looking for ways of diversifying the tools available for them.

Recebido em: 09/2008; Aceito em: 12/2008.

References

BERBER SARDINHA, T. 2002. Metaphor in early applied linguistics writing: A corpus-based analysis of lexis in dissertations. *I Conference on Metaphor in Language and Thought*. Catholic University of São Paulo: Brazil.

BIBER, D., & CONRAD S. 1999. Lexical bundles in conversation and academic prose. In: H. HASSELGARD & S. OKSEFJELL (Eds.) 1999, *Out of corpora - studies in honour of Stig Johansson*. pp. 181-190. Amsterdam/Atlanta,GA: Rodopi.

CAMERON, L. 1999. Identifying and describing metaphor in spoken discourse data. In: L. CAMERON & G. LOW (Eds.), 1999, *Researching and applying metaphor*. Cambridge: Cambridge University Press. pp. 105-132.

_____ 2002. *Metaphor in educational discourse*. London: Continuum.

CHARTERIS-BLACK, J. 2004. *Corpus approaches to critical metaphor analysis*. Basingstoke: Palgrave Macmillan.

DEIGNAN, A. 1999. Corpus-based research into metaphor. In: L. CAMERON & G. LOW (Eds.), 1999, *Researching and applying metaphor*. Cambridge: Cambridge University Press. pp. 203-220.

LAKOFF, G., & JOHNSON, M. 1980. *Metaphors we live by*. Chicago: University of Chicago Press.

_____; ESPENSON, J. & SCHWARTZ, A. 1991. *Master metaphor list: cognitive linguistics group.* Berkeley: University of California at Berkeley. 2nd Edition.

MARTIN, J.H. 1991) *MetaBank: a knowledge-base of metaphoric language conventions.* Unpublished manuscript. Boulder: CO.

MASON, O. 1997. QTAG-a portable probabilistic tagger. Online document, Corpus Research, University of Birmingham: UK. Available at http://www-clg.bham.ac.uk.

MASON, Z. 2004. CorMet: a computational, corpus-based conventional metaphor extraction system. *Computational Linguistics,* **30.**1: 23-44.

PEDERSEN, T., & PATWARDHAN, S. 2002. Distance Perl package (Version 0.1). Duluth: University of Minnesota.

RENOUF, A., & SINCLAIR, J.M. 1991. Collocational frameworks in English. In: K. AIJMER &. B. ALTENBERG. *English Corpus Linguisitcs - studies in honour of Jan Svartvik*. London: Longman. pp. 128-144.

STEEN, G. 1999. Metaphor in discourse: towards a linguistic checklist for metaphor analysis. In: L. CAMERON & G. LOW (Eds.), 1999, *Researching and applying metaphor*. Cambridge: Cambridge University Press. pp. 81-104.

Tony Berber Sardinha is an Associate Professor of Applied Linguistics with the Linguistics Department and the Applied Linguistics Graduate Program, Sao Paulo Catholic University (PUCSP), Brazil. He sits on the Executive Committee of RaAM, Researching and Applying Metaphor, and of ALSFAL, the Latin American Systemic Functional Linguistics Association. He runs a number of websites including the CEPRIL portal for online corpus analysis tools that hosts applications for automatic analysis of a range of linguistic features. He heads several projects, among which are the Brazilian Corpus, a one-billion word online resource, and Br-ICLE, the Brazilian subcorpus of the International Corpus of Learner English. tony@corpuslg.org