An assessment of metaphor retrieval methods

Tony Berber Sardinha[i] (São Paulo Catholic University, Brazil)

1. Introduction

There has been growing interest in using corpora in metaphor research in recent years, and as a result a number of tools and techniques have been proposed and used for metaphor identification. However, very little is known about their ability to retrieve all and only metaphors from corpora. The aim of this paper is report on a quantitative assessment of methods for metaphor retrieval; but it must be stressed that this assessment is not a final evaluation, since performance of any one of these methods may be altered by different test corpora. Out of the many different techniques and instruments reported in the metaphor, corpus linguistics and Natural Language Processing literature, three procedures and three computer tools were selected for assessment.

The procedures are: (1) reading parts of a larger corpus in order to find candidates that are then sought for in the whole corpus through a concordancer; (2) searching for metaphors using different search terms, such as single words, collocates and lexical bundles; and (3) looking for metaphor clusters. The second procedure requires a concordancer, which is a computer tool, but it was classified under the procedures because the point of the section is not to discuss concordancing per se, but the effect of different search term types (generally used in concordancing, but other search instruments are conceivable, such as grep string searches) on metaphor retrieval. Just as with the first procedure, a computer tool of some sort is assumed, but the tool itself is not the focus.

The three computer tools are: (1) finding metaphor candidates through keywords, or words whose frequencies are statistically higher in a corpus than in a comparable reference corpus; (2) finding metaphor candidates through the Metaphor Candidate Identifier, an online tool that looks for metaphorically used words by matching single words and patterns drawn from hand coded training data; and (3) finding metaphor candidates by computing semantic relatedness, more specifically, by computing a measure of the difference in meaning between neighboring words. These tools were chosen because they are free and publicly available[1]. Another tool that has been used in the literature for choosing metaphor candidates is WMatrix (Rayson, 2008), but it requires a paid subscription (even

[1] WordSmith Tools 3.0 is free from Mike Scott's website at www.lexically.net; Kitconc is freely available on José Lopes Moreira Filhos' developer website at www.corpuslg.org/software; the MCI is a free online tool at www2.lael.pucsp.br and www.corpuslg.org/tools; and semantic relatedness is implemented in the free Perl package WordNet::Similarity, available at http://wn-similarity.sourceforge.net.

though a free password for research purposes can be obtained for a limited period of time), and that is why it was not included in this assessment. Other tools such as Cormet (Mason, 2004) and TroFi (Birke, 2005), which are reported in the Natural Language Processing literature, are simply not available for installation. The order of presentation of procedures is from most conventional to least conventional, with partial corpus reading as arguably the most traditional technique, and clustering as the more experimental. For computer tools, the order of presentation is from least demanding to most demanding of computer and programming skills. Keywords is the least demanding because it is implemented in relatively easy to use, point and click programs with graphic interfaces (such as WordSmith Tools and Kitconc). The MCI is much more simple to get started with than either WordSmith Tools or Kitconc, but it is more challenging because it requires some understanding of how it operates under the hood in order for researchers to make sense of its output. And WordNet::Similarity is undoubtedly the most difficult tool to install and operate, as it has no graphic interface and requires programming skills and familiarity with command line interfaces.

Most methods tested here are bottom-up because they are meant to mine corpora for metaphor candidates, rather than seeking predefined candidates. The exception comes under our assessment of search terms for concordancing, which presupposes a set of candidates has already been determined, and therefore may be a case of top-down methodology. As far as the corpus-driven / corpus-based dichotomy (Tognini-Bonelli, 2001), these methods can be either one, because researchers may use them to test particular theories of metaphor, in which case they may be classed as corpus-based, or they may be used to explore how metaphors present themselves lexically in corpora, in which case they may be seen as corpus-driven.

2. Reading portions of the corpus for candidates

One technique commonly employed by metaphor researchers is reading a sample of the corpus texts, noting down any metaphors encountered and then searching the corpus for these. There are a number of questions surrounding this method, motivated by the concern that there might be a substantial number of metaphors left undetected in the corpus because they did not occur in the sample that was read. The main question seems to be then of whether one can retrieve the totality of metaphors from the corpus by reading just a portion of it, and if not, what is the proportion of metaphors retrieved, and whether this proportion rises as the amount of text read increases.

In order to put this technique to the test, different size samples were experimented with. For sample size 1, the texts in the sample were each an individual text (1, 2, 3, etc. up to 14). From then on, each sample size was made up of all possible text combinations for that particular sample size. Therefore, for sample size 2, the texts were pairs of individual texts (1 and 2, 2 and 3, 3 and 4, etc. up to 13 and

14). For sample size 3, the texts were triplets (1, 2, 3; 2, 3, 4; 3, 4, 5; etc.). And so on, until sample size 13, in which case the texts in the sample were a group of 13 texts (1 through 13, 2 through 14, 3 through 14 plus text 1, 4 through 14 plus texts 1 and 2, etc.). These combinations were used in order to prevent bias, which might occur if particular texts were read that had far more metaphor cases than the others. In this way, all texts are considered for reading.

For each of these situations, recall was computed. For this investigation, recall is a measure of the total metaphor types in the corpus retrieved by reading any one sample size. It was computed by dividing the number of metaphor types found in a text portion by the total metaphor types found in the corpus (multiplied by 100). By metaphor type is meant a unique instance of a metaphorically used word; subsequent appearances of the same metaphorically used word were not computed. The higher the recall, the more metaphors were retrieved by reading a particular portion of the corpus. Afterward, the average recall was calculated for the whole sample size. To illustrate, Table 1 shows the figures for text portion 1:

Texts in sample	Metaphors retrieved (A)	Metaphors in corpus (B)	Recall (A/B * 100)
1	123	414	29.7%
2	95	414	22.9%
3	106	414	25.6%
4	134	414	32.4%
5	74	414	17.9%
6	125	414	30.2%
7	95	414	22.9%
8	106	414	25.6%
9	43	414	10.4%
10	105	414	25.4%
11	132	414	31.9%
12	109	414	26.3%
13	64	414	15.5%
14	105	414	25.4%
Average recall for sample size 1			24.4%

Table 1: Recall for text 1

Table 2 shows, for each size sample, the average recall, recall increase and the ratio of recall to sample size (as a percentage). This ratio is a basic measure of effectiveness: the higher the number, the more effective the sample is, in the sense that more metaphors will have been retrieved with less reading input. On the other hand, if the ratio is low (the minimum is 1), then more effort will have been spent by going through a large reading sample to find metaphors.

Sample size	Average recall	Average increase	Recall / sample size

1 (7%)	24.4%	---	3.4
2 (14%)	37.8%	13.3%	2.6
3 (21%)	47.4%	9.7%	2.2
4 (29%)	55.2%	7.7%	1.9
5 (36%)	61.6%	6.4%	1.7
6 (43%)	67.3%	5.7%	1.6
7 (50%)	72.4%	5.1%	1.4
8 (57%)	77.2%	4.8%	1.4
9 (64%)	81.7%	4.5%	1.3
10 (71%)	85.8%	4.1%	1.2
11 (79%)	89.7%	3.9%	1.1
12 (86%)	93.3%	3.7%	1.1
13 (93%)	96.7%	3.4%	1.0
14 (100%)	100%	0%	1.0

Table 2: Recall for reading portions of corpus

These figures show that recall increases as more texts are added to the reading sample, but the increase is not steady: the effect of adding more texts to a smaller sample is more striking than adding to a larger sample. If recall increased at a steady rate, it would increase by 7.1% with each portion (since 100/14 = 7.1). The point of diminishing returns for recall is where the expected average increase drops below 7.1%, which is at sample size 5. This is also the point at which more than half of all the metaphors will have been found. This suggests that a corpus portion consisting of four texts (or 29% of the whole corpus) would be the optimal sample size, beyond which the rate of finding new metaphors would perhaps not justify the effort involved in reading more texts. The effectiveness of the technique, as measured by the ratio recall / sample size decreases as samples get larger. Effectiveness seems to have been compromised after sample size 3, or 21% of the whole corpus, since up to that point the ratio of metaphor retrieval was over 2, meaning twice more metaphors were found than text material was read.

However, these figures show that there are new metaphors in each text, no matter how big a reading sample is. Even a reading sample consisting of all texts but one (13) does not yield all of the metaphors in the corpus.

On the whole, these results indicate that reading a few texts of the corpus for candidates is an effective sampling technique, which enables researchers to retrieve a large portion of the metaphors present across the whole corpus. Reading just 7% (1 text) of the corpus retrieves about a quarter of the metaphors. The majority of the metaphors are found by reading 29% (4 texts) of the corpus.

Again, this conclusion is based on the rationale that researchers will not read an entire corpus in the first place, and that they give some consideration to the amount of text that they will read. The practical advice drawn from these results would then be that researchers should strive to read all of the texts in the their corpus, but if that is not possible (as is often the case with electronic corpora), then they should read at least about 30% of them.

3. Concordancing: Search term choice

Techniques such as the previous one generally presuppose researchers will depend on a concordancer in order to search for the candidates noted during reading. But there are different kinds of search terms that can be used, such as single words, multiple word sequence, and word plus a collocate, to mention a few. The question then arises as to whether different kinds of search words are more reliable than others in retrieving metaphors. In this section, answers to this question will be pursued, but this experiment rests on the assumption that researchers would have an attested set of search terms, obtained for instance by reading portions of the corpus. In other words, the results presented here do not apply to situations in which researchers make up a list of search terms by guesswork, intuition or similar methods.

Different search term types have distinct advantages and disadvantages. Single words are an obviously easy search term to formulate, but they can be ambiguous and therefore retrieve instances of non-metaphor along with metaphors ('waste' would pick up both 'waste our time', which is metaphorical, and 'waste money', which is not). Word sequences, on the other hand, can be trusted to retrieve more unambiguous cases of metaphor ('waste time', 'waste efforts', 'waste our lives', all of which are metaphorical uses of 'waste'), but they can be difficult to formulate, because the exact word sequences that appear in the corpus may be hard to predict. Node and collocate searching may be seen as having the advantages of bundle searching ('waste' followed by 'time' at two words to the right will probably not retrieve any cases of non-metaphor), but it also has about the same drawbacks, namely predicting collocates. Given the problems associated with formulating both bundles and collocations, then it is likely that most researchers would prefer searching their corpora with single words anyway, at least at first, and then probably move on to bundles or collocations, when they have a better idea of the linguistic metaphors in the corpus. But the issue still remains of how reliable single words are as search terms. Less reliable search terms mean extra work for researchers, who will have to read and judge more cases, a situation which may be critical when dealing with large corpora yielding thousands of citations of particular search terms.

The main variable in this investigation is search term type, which is one of the following: single word, bundle, collocation. For bundles, these subtypes were identified, depending on how many words are in the bundle: two words, three words, and four words. For collocations, the following subtypes were recognized, depending on the position of the node: node + 5L (five words to the left of the node), node + 4L (four words to the left of the node), and so on up to node + 5R (five words to the right of the node). The position in which the metaphorically used word occurred did not matter. For bundles, the metaphorically used word(s) could be any of the words comprising the bundle. For collocation, the metaphorically used word(s) could be either the node or the collocate.

The question addressed in this section is how precise each of these search term types is, when used to retrieve metaphors from the corpus. Precision was calculated by dividing the number of metaphors retrieved by the total number of instances retrieved (multiplied by 100). For instance, if a word retrieved 100 citations from the corpus, and 50 of those were metaphors, then precision would be 50% (50 / 100 * 100).

This investigation was carried out as follows. Firstly, all instances of metaphor from one text of the corpus were retrieved and turned into single words, bundles (formed by two, three or four words) and collocations (node plus collocates at positions five, four, three, two and one words to the right and left of the node). These were not mutually exclusive: the same single word was part of a bundle and of a collocation, and collocations of the kinds node + 1L and node + 1R were both two-word bundles. The decision to pull out the search terms from the corpus itself and not to make up the search terms was taken because the intention was not to test our intuition but rather to test the retrieving power of real search terms. If we had made up a list of search terms, some of them might not match any metaphors in the corpus, thus interfering with the results. By drawing the search terms from the corpus, we ensured a level playing field for all search terms, making sure all of them could achieve 100% precision. Secondly, all of these instances were matched against all of their respective metaphor units (single words, bundles and collocations) in the corpus; each time a match was found, a hit was scored. If more than one metaphorically used word occurred in a bundle or collocation, then hits were scored accordingly (a bundle with two metaphorically words received two hits, etc.). Finally, all hits were computed and precision was calculated for each search term type.

Table 3 shows the results for precision for each search word type and subtype.

Search term type	Precision
single word	73.2%
2-word bundle	100%
3-word bundle	100%
4-word bundle	100%
node + 5L	97.9%
node + 4L	97.3%
node + 3L	97.2%
node + 2L	97.7%
node + 1L	100%
node + 1R	100%
node + 2R	98.6%
node + 3R	98.0%
node + 4R	97.0%
node + 5R	97.1%

Table 3: Precision for different search terms

The figures show the most precise search units are fixed word sequences, such as bundles and collocations formed by neighboring collocates, which is not surprising, since fixed patterns normally express a specific meaning. They also show there was no difference among the subtypes of bundles, all of which were 100% precise, unlike collocations, which varied from 97.1% to 100%. The least precise search term type was the single word, as predicted, at 73.2%.

These results suggest a number of interesting findings. Firstly, single words were surprisingly precise, yielding only about one quarter of false positives (non-metaphors instead of metaphors). This is probably due to the fact that this corpus is highly controlled for genre and topic, and so from a probabilistic standpoint, metaphorically used words are generally used to express that one sense only, in a particular phraseology (Berber Sardinha, 2008). With genre, register and/or topic diversified corpora, this figure would probably be lower, as single words take on different meanings in different contexts, expressing a metaphorical use in one context and a non-metaphorical use in another. The practical advice coming from this is that starting with single words is probably a good working strategy for researchers. Later on, as they become acquainted with the phraseology of metaphors in the corpus, they may formulate more precise searches with either bundles or collocations. Secondly, bundle subtypes were equally precise, at 100%, which in practical terms means that with a corpus like this researchers do not need to worry about predicting long fixed word sequences to make precise search searches, as a simple two-word sequence will retrieve metaphors only. This again may be a consequence of the tightly controlled vocabulary used in the corpus, and this is expected to change somewhat with diversified corpora. Overall, these results corroborated Deignan's (2005) findings which indicated that metaphorically used language tends to exhibit a tight phraseology, whereas non-metaphoric language is more freely combining. Finally, there was not much difference among collocate positions, all of which scored above 97%. One might have expected collocates becoming less precise the further away they were from the node, but this was not corroborated here. The practical suggestion arising here is that with corpora like this, researchers should not restrict searches to patterns formed with near collocates, since metaphor phraseology often stretches a long way away from the node.

4. Clustering

Clustering is a property of metaphor distribution in texts, according to which metaphors are distributed unevenly across texts, in such a way that many form groups of metaphorical units occurring near each other. A number of researchers have noticed clustering of metaphors in their analyses of both written and spoken texts. According to Cameron (2008), one of the reasons for clustering is topical, since developing a topic in discourse sometimes requires users to repeat metaphors that are being used to express a particular topic. Another reason for clustering in speech has to do with the tendency for speakers to repeat, reformulate and pick up on each others' points, thus reusing groups of words within a short period of time. To our knowledge, clustering has not been employed so far as a technique for retrieving metaphors. However, it appears as though it could be, perhaps as an awareness raising tool for researchers to apply during metaphor coding. If researchers become aware of clustering, once they spot one case of metaphor in a text or on a concordance line, for instance, they may decide to look more closely for other instances of metaphor nearby. There is no evidence in the literature, though, that this technique has ever been used or whether it is efficient or not. In fact, there is no quantitative evidence of clustering in the literature either, and the aim here is to assess clustering from a quantitative standpoint. A metaphor cluster is defined here simply as an occurrence of two metaphors within a variable stretch of text.

In order to explore clustering quantitatively, the starting point is to assume that there is a textual window around a metaphor where one can find another instance of metaphor, thus creating a cluster. The problem, of course, lies in determining the extent of that window. In this investigation we then look at the issue of finding an optimal window that would allow us to retrieve as many metaphors as possible from the corpus.

The first step was determining a figure that represented the average distance between metaphors in the corpus. This average distance was calculated by dividing the number of word tokens (82881) by the number of metaphor tokens (3800), yielding 21.8, meaning that metaphors are on average about 22 words away from each other. This figure represents the expected distance between metaphors if they were distributed evenly across the corpus. Therefore, a criterion for clustering was established according to which the maximum window size would not exceed the average distance between metaphors across the corpus.

Next, the following window sizes were tested: 5, 10, 15 and 20 words, and recall was calculated for each window size. Recall was computed for each text by dividing the number of metaphor tokens occurring within the window by the total metaphor tokens in the text multiplied by 100. Finally, mean recall for each window size was computed by averaging out the individual recall figures for each text. To illustrate, Table 4 below shows the results for window size = 5.

Text	Metaphor tokens within window	Metaphor tokens in text	Retrieval
1	106	395.0	26.84%
2	68	254.0	26.77%
3	65	274.0	23.72%
4	77	383.0	20.10%
5	27	210.0	12.86%
6	85	357.0	23.81%
7	59	285.0	20.70%
8	62	256.0	24.22%
9	12	75.0	16.00%
10	50	293.0	17.06%
11	63	289.0	21.80%
12	62	308.0	20.13%
13	28	133.0	21.05%
14	66	288.0	22.92%
Average			21.28%

Table 4: Clustering retrieval for window size = 5

Results indicate that with a window of size 5, an average 21% of the metaphors fall within a cluster. This was repeated with the other window sizes, and the results appear in Table 5.

Window size	Average recall
5	21.28%
10	41.14%
15	55.83%
20	65.25%

Table 5: Clustering recall

The table presents a couple of interesting findings. The first is that, as would be expected, recall rises as the window size expands. Wider windows pick up more metaphors, whereas narrower windows miss out on more metaphors. The second is that none of the window sizes returned recall rates near 100%; even at a generous window size such as 20, which is near the even distribution (22), recall is only about 2/3 of all metaphors. With a window size this wide, there is not much point in looking for metaphors within clusters, as they would be so large that there would be very few gaps between them, thus essentially forcing researchers to read the whole corpus.

The practical advice that could gleaned from this would be to stick to narrow window sizes such as 5 and 10, which, in corpora similar to ours, would help retrieve up to 40% of the metaphors. In addition, window sizes such as these normally fit within the length of most concordances. This may enable researchers to spot metaphors in the vicinity of node words.

5. WordSmith Tools Keywords

Keywords are words whose frequencies are statistically higher in a corpus in comparison to a reference corpus. Keywords is also the name of an application that is part of the corpus analysis package WordSmith Tools (Scott, 1997), which extracts keywords automatically. Keywords can be extracted by a number of different tools besides WordSmith Tools, including Kitconc (Moreira Filho, 2008) and the CEPRIL Keyword Tool (www2.lael.pucsp.br/corpora). Keywords have been used in metaphor research (Berber Sardinha, 2009; Partington, 2006; Philip, 2008) for the general purpose of selecting candidates for close inspection.

As with the other techniques, there are questions surrounding the reliability of keywords as a means of metaphor retrieval, not least because little is know about the relationship between metaphor and outstanding lexical frequency, the guiding principle behind keywords. The specific aim here is to find out what proportion of metaphors can be retrieved through keyword extraction, and how precise this method is. These seem important issues surrounding keywords, even if researchers employ keywords for purposes other than retrieving the most metaphors out of their corpora.

To investigate this issue, the following procedures were followed. Firstly, the keywords of the corpus were extracted in WordSmith Tools version 3, by comparing the word frequency of the corpus to that of the Banco do Português (version 1), a large register-diversified corpus of Brazilian Portuguese comprising over 230 million words of spoken and written language. The settings for keywords were as follows: max keywords 500000, max p. value .05, keywords procedure log-likelihood. These settings enabled all keywords to be extracted, and not just the default 500. A total of 2532 keywords were produced, including both positive and negative ones. Secondly, all metaphorically used words in the corpus were listed. Thirdly, positive keywords were separated from negative words. Positive keywords are the usual keywords, that is, each keyword has a frequency statistically higher than its frequency in the reference corpus; negative words are the reverse of these, in the sense that their frequencies are statistically lower than in the reference corpus. Negative keywords, if available in a particular corpus, appear in red at the bottom of the screen in WordSmith Tools version 3. The keyword lists were split into samples that started with the top 100 keywords and were incremented by 100 keywords; samples were then 1000, 2000, 3000 and so on up to 2044 for the positive keywords and up to 488 for the negative ones. Finally, these metaphorically used words were then matched against the keywords, the number of exact matches was recorded, and performance metrics were computed (precision and recall). Precision was calculated by dividing the total matches for a particular sample by the size of that sample; recall was computed by dividing the total matches for a particular sample by the total metaphorically used words in the corpus (414).

Results for positive keywords appear in Table 6.

Sample	Matches	Precision	Recall
100	7	7%	2%
200	16	8%	4%
300	26	9%	6%
400	38	10%	9%
500 (default)	46	9%	11%
600	64	11%	15%
700	70	10%	17%
800	75	9%	18%
900	85	9%	21%
1000	88	9%	21%
1100	92	8%	22%
1200	103	9%	25%
1300	110	8%	27%
1400	119	9%	29%
1500	125	8%	30%
1600	129	8%	31%
1700	142	8%	34%
1800	147	8%	36%
1900	160	8%	39%
2000	170	9%	41%
Whole list (2044)	172	8%	42%

Table 6: Precision and recall for positive keywords

As can be seen in the table, the best precision score was for the 600 keyword sample, which amounts to 29% of the keyword output. The best recall mark was for the whole list, with 42% of the total metaphors retrieved.

Results for negative keywords appear in Table 7.

Sample	Matches	Precision	Recall
100	8	8%	2%

200	13	7%	3%
300	19	6%	5%
400	23	6%	6%
500	24	5%	6%
Whole list (588)	24	4%	6%

Table 7: Precision and recall for negative keywords

The results show the best precision score is for the top 100 negative keywords, at 8%, and the best recall is for the whole list, at 6%. Topmost negative keywords are those bearing the most marked frequencies, meaning they are the most rare words in the corpus. This suggests some metaphorically used words are unusual in the corpus.

Table 8 shows the overall recall achieved by the Keywords procedure.

Total metaphors retrieved	Whole list		Portion of list			
			Corresponding to highest precision		Default 500 keywords	
By positive keywords	172	42%	64	15%	46	11%
By negative keywords	24	6%	8	2%	---	---
Total	196	47%	72	17%	46	11%

Table 8: Overall recall by Keywords

The Keywords procedure retrieved less than half of the metaphors, if we include both positive and negative keywords. About 53% of the metaphorically used words were not keywords at all, that is, their frequency was statistically similar to their frequency in the comparison corpus. This suggests metaphorically used words are often not particularly frequent or rare, otherwise they would have made keywords, positive or negative. The highest recall was reached with the whole list of keywords (including positive and negative), but it would be unusual for researchers to consider the full list of keywords in their analysis, not least because the list extracted here was obtained with the least stringent criteria possible for keyword extraction in WordSmith Tools. Normally, researchers use the default criteria which produces a 500-keyword list, and for that list, recall was only 11%. Recall for the point on the list where precision was highest was slightly better at 15%, but in practice such a point is hard if not impossible to determine, given that researchers will not know which keywords are metaphorically used by the time they run Keywords.

In conclusion, keywords do not seem to be a particularly effective retrieval technique, at least with the data used here. That does not mean, however, that selecting words with keyword status is not relevant for metaphor research. The fact that words have a marked frequency may be important in a number of ways, as pointed out in the literature, given the fact that keywords may signal important textual properties such as aboutness, style and textual salience, among other attributes, all of which may be relevant to particular metaphor research projects. These findings pertain to metaphor retrieval only, and not to the relevance of keywords per se. One further point is that we must remind ourselves Keywords was not designed to retrieve metaphors, and therefore it cannot be criticized for not doing particularly well at a job it was not intended to do.

6. Metaphor Candidate Identifier

The Metaphor Candidate Identifier (MCI) is a computer program developed by Berber Sardinha (2007) which aims specifically at retrieving metaphorically used words from corpora. It works by matching each word in a corpus, its patterns and its part of speech to a set of five databases, and then calculating the average probability of that word being metaphorically used. These databases were compiled from hand-coded concordances (the ‘training data’), where each node word was judged as metaphorical or not based on principles similar to those proposed in the MIP. Each database holds specific information about single words, 3-word bundles preceding and following each word, the immediate collocates to the left and right of the word (called ‘framework’) and the part of speech assigned to that word by a tagger (Tree-Tagger). The output of the program is an ordered list of candidate words, sorted by its probability of metaphorical use. The MCI is an online tool that is available in two versions, one for analyzing Portuguese corpora and another for English corpora; both versions can be accessed for free on the web at the CEPRIL (Center for Research, Resources and Information on Language, Sao Paulo Catholic University) website at www2.lael.pucsp.br.

To illustrate how the program goes about identifying metaphor candidates, let’s take the following sentence from the ‘Ozone’ text in Cameron (2003 :168), where ‘made’ is a metaphorically used word:

‘But not all the energy made by the Sun is safe.’

The MCI would check each word in that sentence, and for ‘made’, processing would be carried out in the following way:

- made:
    - Check single word database, which stores each word that was found to be metaphorically used in the training data, together with its probability of metaphor use. ‘Made’ is found on the database, with a probability of .6000. This value is stored in the program’s memory. If this word were not on the database, this would mean it was never found in the previously hand-coded texts to be metaphorically used, either because it appeared in the training data in its basic sense or it never appeared at all in the texts. Either way,

the program would store the value of .00001 for it.

- Extract 3-word bundle preceding it: ‘all the energy’
- Check that bundle in the ‘left bundle database’, which stores all 3-word bundles that preceded each metaphorically used word in the training data, together with its probability of metaphor use. The bundle is not found there, and so the program stores the value of .00001 for it.
- Extract 3-word bundle following it: ‘by the Sun’
- Check that bundle in the ‘right bundle database’, which stores all 3-word bundles that followed each metaphorically used word in the training data, together with its probability of metaphor use. The bundle is not found there either, and so the program stores the value of .00001 for it.
- Extract the lexical framework around it: ‘energy … by’.
- Check that bundle in the ‘framework database’, which stores such patterns occurring around each metaphorically used word in the training data, together with its probability of metaphor use. The framework is not found there either, and so the program stores the value of .00001 for it.
- Assign a part of speech to it: verb
- Check the probability for verbs in the ‘part of speech database’, which stores the probability of each part of speech being metaphorically used in the training data. The probability for verbs is .2061.
- Average out these probabilities and assign this value to the word: .1612.

This score of .1612 is low, given that final scores can range from .0001 to 1[ii]. But it is not the absolute score that matters, but its place in the rank of scores. As it turns out, this was the 7th highest ranking score for that particular text, and therefore it is likely that this word would have been considered for analysis.

Tony Berber Sardinha 6/17/09 10:53 PM
Comment: Repeated: but ... but

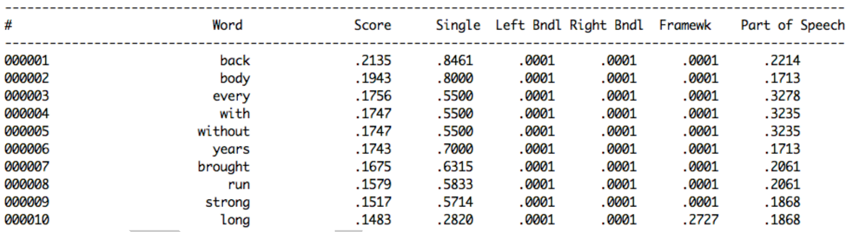
In order to evaluate the performance of the MCI, I first looked for texts or corpora that had been previously coded for metaphor use, so that the analysis were independent and did not necessarily reflect my own, but found only two texts, both in Cameron (2003). This shortage of publicly available datasets highlights the difficulties involved in producing and testing programs to detect metaphors, because it leaves it to developers to find ways to hand tag their own corpora.

In addition to these two texts, three others were added to the test sample, resulting in a five-text sample. After they were hand analyzed, they were submitted one by one to the MCI, and finally the computer and the hand analyses were compared. The five texts were the following: (1) Atmosphere text: a text about the Earth’s atmosphere, from a book on the ozone layer, included in Cameron (2003); (2) Heart text: a text on the human heart, from a book on the human body, again included in Cameron (2003); (3) Obama text: a news story on the US President Barack Obama's visit to the Middle East, published on the Boston Globe, accessed on Google News; (4) Fed text: A news story on the efforts by the US Federal Reserve to end the recession, posted by Reuters, accessed on Google News; (5) Lobby text: A news story about left-wing lobbying groups in Washington, DC, published by the Washington Post, accessed on Google News. The first two texts were chosen because they had already been analyzed for metaphor in a major publication in the field. The other texts were picked at random to complete a 5-text sample. Table 9 displays the length of each text in (valid) word tokens and metaphorically used words.

Text	Tokens	Metaphorically used word types
Atmosphere	120	16
Heart	122	19
Obama	268	38
Fed	218	34
Court	585	74

Table 9: MCI test corpus

Figure 1 is a snippet of the MCI output for the heart text.



```
-----------------------------------------------------------------------------------------------
#                          Word            Score      Single  Left Bndl Right Bndl  Framewk    Part of Speech
-----------------------------------------------------------------------------------------------
000001                     back            .2135      .8461     .0001     .0001      .0001      .2214
000002                     body            .1943      .8000     .0001     .0001      .0001      .1713
000003                    every            .1756      .5500     .0001     .0001      .0001      .3278
000004                     with            .1747      .5500     .0001     .0001      .0001      .3235
000005                  without            .1747      .5500     .0001     .0001      .0001      .3235
000006                    years            .1743      .7000     .0001     .0001      .0001      .1713
000007                  brought            .1675      .6315     .0001     .0001      .0001      .2061
000008                      run            .1579      .5833     .0001     .0001      .0001      .2061
000009                   strong            .1517      .5714     .0001     .0001      .0001      .1868
000010                     long            .1483      .2820     .0001     .0001      .2727      .1868
```

Figure 1: Portion of MCI output

As said above, the MCI analyzed all of the words in each text, and therefore if an analyst were to check each word in its output, he/she would end up finding all of the metaphors in the text. But that is not the point of the MCI: the program was conceived of as a tool for screening texts and bringing to sharp relief the most likely metaphorically used words, which appear at the top of the output; the further away from the top, the least likely it is a word will be a metaphor. Consequently, it is not wise to consider the whole output for each text, but just samples of the top n words in the output.

This was taken into consideration in this evaluation of the MCI. Recall was the number of correctly identified metaphorically used words in the output sample as a proportion of the total metaphors in the text, where correct means matching the human analysis. Precision, in turn, was measured by dividing the number of correctly identified metaphorically used words by the number of words in the output sample.

To illustrate, in Table 10 is a portion of the output for the heart text, with an extra column indicating if the candidate is indeed a metaphorically used word:

Output sample size	Candidate	Score	True metaphorically used word?
000001	back	.2135	Yes
000002	body	.1943	No
000003	every	.1756	No
000004	with	.1747	No
000005	without	.1747	No
000006	years	.1743	No
000007	brought	.1675	Yes

Table 10: Portion of MCI output

Recall: In this text, there are 19 metaphorically used words. With output sample size = 1, the only candidate is 'back', which is correctly identified as metaphorically used. Hence, recall = 1/19 = 5.3%. With sample size = 2, there are two candidates, but only 1 is a true metaphor, and so recall is still 1/19 = 5.3%. Finally, with sample size = 7, recall is 2/19, or 10.5%.

Precision: With sample output = 1, only one candidate was offered by the program, and this single candidate is a true metaphorically used word, hence precision = 1/1 = 100%. With sample output = 2, two candidates are offered, but only one is correctly identified, therefore precision = ½ = 50%. Finally, with sample output = 7, 7 candidates were offered, but only two were correct, hence precision = 2/7 = 28.6%.

The results for precision appear in Table 11. Results are presented for output sample sizes from 10 to 50 and for an output sample size that captured the majority of the metaphors in each text (identified as Recall > 50%).

	Output sample size					
Text	10 candidates	20 candidates	30 candidates	40 candidates	50 candidates	Recall > 50%
Atmosphere	14%	15%	17%	17%	12%	13%
Heart	29%	20%	20%	15%	12%	12%
Obama	78%	61%	52%	42%	36%	30%
Fed	43%	33%	30%	30%	27%	19%
Lobby	30%	30%	30%	26%	31%	16%
Average	*39%*	*32%*	*30%*	*26%*	*24%*	*18%*

Table 11: MCI precision

The results indicate that precision varies considerably across texts, ranging from 12% to 78%. The text on which MCI performed best was the Obama text, the likely reason being that the vocabulary and phraseology of this text must be more familiar to MCI than the other texts, as its words in patterns are perhaps more similar to the training data. Performance also varies with output sample size, with bigger output sizes yielding generally lower precision values, and this is because with larger output samples there are more opportunities for the program to suggest unsuccessful candidates, as there are more non-metaphors than metaphors in each text. Overall, the average precision of the MCI ranges from 18% to 39%.

The results for recall are shown in Table 12. The central columns in the table show what size output sample is required to achieve 25%, 50%, 75% and full recall of the metaphorically used words. Sample size is shown as the number of candidates indicated in a portion of the MCI output (starting at the top of the output) and as a percentage of the full output length; for instance, for the atmosphere text, 25% recall is achieved with 21 candidates, taken from the top of the list, and these 21 candidates represent 18% of the total output of 120 tokens.

Tony Berber Sardinha 6/17/09 11:33 AM
Comment: Recall is dependent on the size of the output. The full output will reach 100% recall, simply because all candidates will be taken into account. A longer output is bound to achieve higher recall, but it is less likely to be considered by researchers as a helpful subsidy for analysis. Consequently, the question is what size output is needed to achieve particular recall levels.

Text	Output sample needed to reach recall level, starting at the top of the output								Full output length in tokens
	25% recall		50% recall		75% recall		100% recall		
Atmosphere	21	18%	69	58%	77	64%	111	93%	120
Heart	33	27%	82	67%	108	89%	116	95%	122
Obama	17	6%	52	19%	150	56%	248	93%	268
Fed	30	14%	76	35%	150	69%	205	94%	218
Lobby	64	11%	236	40%	358	61%	579	99%	585
Average	*33*	*13%*	*103*	*39%*	*169*	*64%*	*252*	*96%*	*263*

Table 12: MCI recall

As can be seen, recall figures vary according to text. As with precision, the Obama text is the one where the MCI did best; compared to the other texts, it is easier to retrieve a larger proportion of metaphors in this text than in the others. The explanation for this is the same as for precision, that is, this text must contain more of the single words and patterns that MCI has been trained to recognize. On average, in order to retrieve ¼ of all metaphors (25% recall), researchers would need to analyze the top 13% of the output list; in order to retrieve ½ of the metaphors (50% recall), they would have to look at three times as much output (39%); for 75% recall, 64% of the output must be considered, and for full recall, almost all of the output (96%). The best performance is for 25% recall, because analysts can achieve this with about half of that length of the output (13%); for higher recall, analysts must tackle longer portions of the output; full recall can generally only be achieved by analyzing the whole output.

These figures suggest the MCI is moderately reliable, with precision ranging from 18% to 39% on average, and 50% recall achieved by analyzing about 39% of the list of metaphor candidates provided by the tool. These figures may seem disappointing, but if we take into consideration the fact that human metaphor analysts also disagree a great deal among themselves, then they do not look so frustrating. Cameron (2003 :169) reports rater decisions for the Ozone text (the 'atmosphere' text analyzed here) which indicates that the 25 raters working on that text agreed all the time on just two of the 14 metaphors in the text (2/14 = 14%). Beigman Klebanov, Beigman and Diermeier (2008) calculated inter-rater agreement across nine annotators, working on metaphor identification in 2364 newspaper paragraphs, and found out that agreement was between 1.7% to 4%. Such levels of disagreement can be lowered by having discussion sessions among raters, in which they discuss differences in coding and try to reach a consensus (Pragglejaz Group, 2007). MCI precision, at 18% to 39%, exceeds these figures for human raters working on the same texts. It is probably unwise to compare results on such different tasks, but the point is that it is unrealistic to expect human analysis to be totally consistent in their judgment of metaphor, just as it is not viable to expect machine identification of metaphor to be fully reliable.

7. Semantic relatedness

Semantic relatedness means the degree of 'closeness' in meaning between two or more words. For instance, 'elephant' and 'violin' may be considered 'distant' in meaning because, among other reasons, one word refers to a large animal, the other to a small musical instrument; on the other hand, 'cat' and 'dog' may be seen as 'closer' in meaning, because both refer to animals that share several characteristics, among which the fact that they are mammals, furry, and are normally raised as pets.

Semantic relatedness has been explored as a metaphor identification tool by Berber Sardinha (2007), who suggested some metaphors could be spotted in corpora via a specific program which would automatically assign a relatedness value for each word pair in the texts. The rationale was that word pairs formed by a metaphorically used word and a non-metaphorically used word would show low scores for relatedness, compared to other word pairs in the corpus. Such low relatedness scores would in turn be a reflection of incongruity, which underlies linguistic metaphor. As Cameron (2003 :9) explains:

'the linguistic presence of metaphor is signaled by a lexical item that can have an interpretation which is incongruous with the discourse context, or with the meaning created by the co-text. [For example, in], 'the atmosphere is a blanket of gases', the lexical item 'blanket' links to a different semantic field or conceptual domain from that intimated by (...) 'gases' (...). The lexical item 'blanket' is the focus of the metaphor, or the Vehicle term, and the rest of the phrase of sentence, against which it appears incongruous, is called the frame of the metaphor (Black, 1979)'

Semantic relatedness can be implemented in many different ways, but the best know software for this is perhaps WordNet::Similarity, developed by Pedersen and Patwardhan (2006), which is actually a Perl package that uses WordNet, a lexical database, to compute the semantic similarity between pairs of words. WordNet is an electronic lexicographic database, containing thousands of words and their definitions hierarchically structured. The actual computation of semantic relatedness is carried out by a range of different methods, identified by acronyms such as lch (after Leacock and Chodorow, the proponents of one such method), resnik, lin and lesk. Each method uses a different algorithm, but all of them are based on the basic idea that each word input to them is searched for on the WordNet lexical database, its position on the database is stored, the positions for both words are compared, and a score is given to represent how close these positions are. Words that are semantically related tend to appear closer to each other in the WordNet hierarchy than words that are unrelated.

Since my aim with this technique was in a sense to try and reproduce human judgment during metaphor analysis, then I decided to choose a relatedness measure that also approximated to human judgment in semantic relatedness tasks. Luckily, semantic relatedness measures have been tested empirically for their ability to match the judgments made by human raters evaluating relatedness between words. Seco, Veale and Hayes (2004) calculated the semantic relatedness for a list of noun pairs that had been rated by humans in Miller and Charles (1991), and found out that the measure which had the strongest correlation with the raters' judgments was Leacock and Chodorow (.82 correlation). Consequently, Leacock and Chodorow was selected as the measure for relatedness for this investigation. Interestingly, Resnik (1995) replicated Miller's and Charles's experiment and found that his group of raters did not agree 100% with the previous one, rather they correlated at .89. Warin, Oxhamar and Volk (2005) consider this to be the upper-bound for a computer program to achieve, meaning that the best we can realistically expect from an some software analyzing the relatedness between words is that it match human analysts 89% of the time. This level was also taken to be the highest possible level for the metaphor identification trial carried out here.

Returning to the example above of 'blanket' and 'gases', I made up a test list of word pairs, consisting of this incongruous pair and of two more congruous pairs deriving from it, namely 'blanket' and 'bed', and 'gases' and 'oxygen', both consisting of words that are intuitively related. I then ran the list through similarity.pl with measure lch, and got the following results (decimals rounded off):

blanket gases 1.5

blanket bed 3.0

gases oxygen 3.0

As can be seen, the incongruous pair received the lowest relatedness score, out of a maximum possible of 3.7.

Next, I ran a larger test on a set of 7,524 concordance lines from the BNC that I had previously hand coded for metaphor, following the basic principles laid out in the MIP (Pragglejaz Group, 2007). This set was built from a selection of search words taken from the top 500 most frequent words in the BNC. Each concordance line node was coded as either metaphorically or non-metaphorically used with a tag. The aim here was to verify to what extent semantic relatedness would capture these metaphorical node words, more specifically by running a Unix shell script through the corpus that did the following, for each concordance line: (1) clip the node word and the word occurring at position 1R (one word to its right) and produce a word pair, then do the same with word occurring at position 2R and produce a second word pair, so that, for instance, for search word 'bank' on a concordance line such as 'the bank invested millions' the resulting word pairs would be 'bank invested' (bank + 1R) and 'bank millions' (bank + 2R); (2) compute relatedness for each of these word pairs. The reason for restricting the span to positions 1R and 2R was that it would be enough to retrieve cases such as 'waste time' (waste + 1R) and 'blanket of gases' (blanket + 2R). Other cases where the incongruous pair is further removed than 2R ('blanket of deadly gases', 'waste a lot of our time', etc.) were not picked up, and this is a limitation of the procedure. The aim here was to try it out and see if it looks promising, and if it does, extend it to capture a wider span of collocates in further research. It must be stressed that having concordance lines is not a requirement for running this procedure; in fact, similarity.pl is meant to be used with regular running text.

Relatedness was computed with Leacock Chodorow, with the 'all senses' option activated, which makes similarity.pl display all of the senses related to the words in the pair that are featured in WordNet. This was necessary because the exact WordNet sense of every word in the word pair set was not determined. In order for that to be possible, the whole corpus should have been sense disambiguated, which was not feasible at the time. But even if it had been, there is no guarantee that the disambiguation would have been perfect, and the remaining problems would have to be hand corrected, which may be more time consuming that hand coding the texts for metaphors in the first place. The side-effect of the 'all senses' option was that it multiplied the number of word pairs. Hence, instead of 12,055 unique word pairs, the word pair count was 343,347!

After the calculation of relatedness was completed, a second script went through the output, sorted it by similarity score in reverse order (with the least related pairs at the top), and counted how many word pairs contained a metaphorically used word in samples of the most unrelated word pairs. Results appear in Table 13.

Sample of least related word pairs	Pairs containing a metaphorically used word		Pairs not containing a metaphorically used word			
	Total	Unique metaphors retrieved	Total	Unique false positives retrieved	Precision	Recall
Top 1,000 (.3%)	1,000	7 (4%)	0	0	100.0%	3.9%
Top 10,000 (3%)	10,000	41 (23%)	0	0	100.0%	22.8%
Top 50,000 (15%)	41,152	154 (86%)	8,848	27 (8%)	85.1%	85.6%
Top 100,000 (29%)	41,152	154 (86%)	58,848	134 (39%)	53.5%	85.6%
Top 150,000 (44%)	41,152	154 (85%)	108,848	259 (75%)	37.3%	85.6%
Top 200,000 (58%)	46,376	177 (98%)	153,624	340 (98%)	34.2%	98.3%
All 343,437 (100%)	76,593	180 (100%)	266,844	346 (100%)	34.2%	100.0%

Table 13: Semantic relatedness recall

The results show that the most unrelated word pairs retrieve metaphors only, which in turn underscores the usefulness of this technique in metaphor retrieval. There were only metaphorically used words among the top 10,000 least related pairs, yielding 100% precision. As we go down the list, though, non-metaphors begin to crop up, reducing precision gradually down to 34% when all pairs are considered. However, the number of actual metaphorically used word types retrieved is very small compared to the total word pairs; for a 1K word pair sample, only 7 unique metaphorically used word types are retrieved, and for a 10K pair sample, only 41 distinct cases of metaphor are retrieved. This was caused by the multiplication of word pairs triggered by the 'all senses' option, as mentioned above. This in turn affects recall, which is very low with smaller samples and gradually improves as more word pairs are taken into account.

The best scenario seems to be with a 15% sample of the output, which reveals about 85% of the metaphors in the corpus with 86% precision. This is very close to the upper-bound of 89% suggested by Warin, Oxhamar and Volk (2005) for semantic relatedness tasks.

In conclusion, this technique seems to have some potential for metaphor retrieval, in that it appears to tap into incongruity, an important feature in metaphor deployment and interpretation. Incongruity seems to be manifested to a certain degree by the use of semantically unrelated word pairs near each other in text. However, more research is needed before it can ascertained that this is a reliable tool for metaphor detection. One aspect of the output that called my attention was the fact that the lowest scoring word pairs received a score of - 1000000, which is assigned to comparisons of words of different parts of speech, such as 'case' (noun) and 'is' (verb). WordNet::Similarity does not 'cross part of speech boundaries', and so whenever such a pair is submitted to it, it gives a warning and assigns this lowest relatedness value to the pair. Interestingly, all such cases involved metaphorically used words. More trials are needed, then, to see if this is a regular feature with metaphors and semantic relatedness. Apart from this word of caution, the practical suggestion that can be drawn from this exploration is that researchers may use this technique to retrieve some candidates for analysis, but in order to do so they will need to learn some fairly advanced programming skills in Shell and Perl. These are needed for a range of tasks from simply installing the Perl WordNet::Similarity package to writing scripts for preparing the word pairs, submitting them to the package, and handling its lengthy output (depending on the corpus in question).

8. Conclusion

To summarize, here is the best performance indicators of each procedure and tool assessed here:

Tony Berber Sardinha 6/17/09 10:52 PM
Comment: are

- Reading a portion of the corpus for candidates: 30% of the texts yielded more than 50% recall.
- Concordance search terms: fixed word sequences achieve 100% precision, whereas single words reach 73% precision.

- Looking for metaphors in clusters: a span of 5 to 10 words around metaphors retrieves about 40% of the metaphors.
- Keywords: 9% precision and 11% recall for the default listing, or 47% recall for the longest possible listing.
- MCI: up to 39% precision, and 50% recall when 39% of the candidates are considered.
- Semantic relatedness: 85% recall and 86% precision with a 15% sample of the output.

According to these figures, the most reliable procedure is using lexical bundles as search terms for concordancing, and the least reliable procedure is clustering. As far as the tools, the most reliable one is semantic relatedness, whereas the least reliable is Keywords.

Nevertheless, each technique has its own merits and demerits, which must be pointed out in relation to other techniques. Reading corpus portions is fine for small corpora, but with larger corpora a reading portion of 21% may translate into a portion that is too big to handle. For instance, for a 10 million corpora, this would mean a 2.1 million word reading load. Likewise, concordancing with bundles is great, but it presupposes one has reliable search terms to start with. If getting these terms depends on reading large portions of a corpus, then one may not be able to obtain the reliable terms in the first place. Clustering was shown to work with larger windows, but putting this idea in practice may be a bit confusing, as researchers would have to look further and further away from each metaphor to find other metaphors. In addition, many of these windows will partially overlap, and as they add up, they will cover larger and larger stretches of text, which in turn may defeat efficiency. Keywords are relatively easy to obtain, but they are not particularly reliable and require a reference corpus, which may not be available for particular languages. The MCI has a simple interface, and it’s the only one (in this study) dedicated exclusively to metaphor retrieval, but it is only available for English and Portuguese. WordNet::Similarity proved to hold some promise for metaphor detection, but metaphor researchers in Humanities departments may find it hard to install and operate, and it works with English data only.

In general, procedures that depend on reading a corpus for candidates may break down with larger corpora, and this is where automatic retrieval techniques can come in and prove their worth. Researchers may have to forego some reliability by using automatic retrieval methods, but a loss of reliability is made up by a gain in volume, as larger corpora can provide a richer variety of metaphor instances than smaller ones.

To conclude, the choice of method will always depend on the aims of particular research projects and thus it does not make sense to impose ‘the best method’, but researchers must bear in mind the advantages and drawbacks of the methods they employ for metaphor retrieval, and if possible use more than one method to improve efficiency and coverage.

References

Beigman Klebanov, Beata, Beigman, Eyal, & Diermeier, Daniel. (2008). Analyzing disagreements. In Ron Artstein, Gemma Boleda, Frank Weller & Sabine Schulte im Walde (Eds.), Proceedings of Workshop on Human Judgements in Computational Linguistics, Coling 2008, (pp. 2-7). Manchester, UK.

Berber Sardinha, T. (2007). Finding metaphors with the help of the computer. Workshop presented at the RaAM Workshop 'Issues in Researching Metaphor in Discourse', Universidad de Castilla - La Mancha, Spain, 22-23 March 2007.

Berber Sardinha, T. (2008). Metaphor probabilities in corpora. In Mara Sofia Zanotto, Lynne Cameron & Marida Cavalcanti (Eds.), *Confronting Metaphor in Use: An Applied Linguistic Approach* (pp. 127-148). Amsterdam/Atlanta, GA: Benjamins.

Berber Sardinha, T. (2009). *Pesquisa em Lingüística de Corpus com WordSmith Tools*. Campinas: Mercado de Letras.

Birke, Julia. (2005). A Clustering Approach for the Unsupervised Recognition of Nonliteral Language. M.Sc., Simon Fraser University.

Black, M. (1979). More about metaphor. In A. Ortony (Ed.), *Metaphor and thought* (pp. 19-43). New York: Cambridge University Press.

Cameron, L. (2008). Metaphor and talk. In R. W. Gibbs (Ed.), *The Cambridge Handbook of Metaphor and Thought* (pp. 197-211). New York: Cambridge University Press.

Cameron, Lynne. (2003). *Metaphor in Educational Discourse*. London: Continuum.

Deignan, A. (2005). *Metaphor and Corpus Linguistics*. Amsterdam/Philadelphia: John Benjamins.

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

Miller, George, & Charles, WG. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1-28.

Moreira Filho, José Lopes. (2008). Kitconc Computer Software. (Version 1). São Paulo: Corpuslg.org, available at www.corpuslg.org/software.

Partington, Alan. (2006). Metaphors, motifs and similes across discourse types: Corpus-assisted discourse studies (CADS) at work. In A. Stefanowitsch & S.T. Gries (Eds.), *Corpus-based Approaches to Metaphor and Metonymy* (pp. 267-304). Berlin; New York: M. de Gruyter.

Patwardhan, S., & Pedersen, T. (2006). Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts. In Proceedings of Twentieth National Conference on Artificial Intelligence, (pp. 1692-1693). Pittsburgh, PA.

Philip, Gill. (2008). Metaphorical keyness in specialised corpora. Unpublished manuscript. Available at http://amsacta.cib.unibo.it.

Pragglejaz Group. (2007). MIP: A Method for Identifying Metaphorically Used Words in Discourse. *Metaphor and Symbol, 22*(1), 1-39.

Rayson, Paul. (2008). *Wmatrix: a web-based corpus processing environment*. Computing Dept., Lancaster: Lancaster University.

Resnik, Philip. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of 14th International Joint Conference on Artificial Intelligence, (pp. 448-453). Montreal, Canada.

Scott, Mike. (1997). WordSmith Tools. Version 3. Computer Software. Oxford: Oxford University Press, available at.

Seco, Nuno, Veale, Tony, & Hayes, Jer. (2004). An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In Proceedings of ECAI2004, the 16th European Conference on Artificial Intelligence. Valencia, Spain.

Tognini-Bonelli, E. (2001). *Corpus Linguistics at Work*. Amsterdam/Atlanta,GA: John Benjamins.

Warin, Martin, Oxhammar, Henrik, & Volk, Martin. (2005). Enriching an Ontology with WordNet based on Similarity Measures. In Proceedings of MEANING-2005 Workshop. Trento, Italy.

[i] The author is grateful to CNPq (Brasília, Brazil) for grants # 307307/2006-9, 400574/2007-1, 450239/2006-3, 350455/2003-1, and Capes for grant # 0397/04-0, as well as the Researching and Applying Metaphor (RaAM) International Association for their support.

[ii] Averaging out probabilities is not the right way to determine the joint probability of words being metaphorically used given the individual probabilities determined by the program, hence the final score is not meant to be an accurate representation of its actual joint probability, but simply a figure that represents the average score obtained by a particular word. This score is needed for ranking the word in the program output, which is sorted in reverse order, with the highest scoring words on the top. In other words, the final score should not be interpreted as a true probability, but is simply a means for ranking the words in the output.