International Journal of Knowledge and Language Processing
Volume 3, Number 1, **2012**

www.ijklp.org
KLP International ©2012 ISSN 2191-2734
pp.15—27

Identifying English Verb Metaphors Using Statistical and Clustering Approaches

Jian Xu Jingsong Yu*
School of Software and Microelectronics
Peking University
HaiDian District, Beijing, China
wyatt198258@126.com; *yjs@ss.pku.edu.cn

Received March 2011; revised October 2011

ABSTRACT. *In this paper we propose statistical and clustering approaches to automatically identify English verb metaphors. Our approaches need the WordNet resource to compute affinity and similarity between words. Using TroFi Example Base as the base metaphoric resources, we caught all the verbal targets by computing similarity between clustered results and the base metaphoric resources. We then present the experimental results, discovering that the clustering approach increases the recall and precision achieved simply by the statistical approach.*

Keywords: Verb Metaphors, Clustering, affinity and similarity

1. **Introduction.** Metaphors are pervasive in our language and the essence of them is understanding and experiencing one kind of thing in terms of another (Lakoff and Johnson, 1980). In example (1):

(1) "My head is like a coreless apple."

The tenor 'head' and the vehicle 'apple' are different kinds of things. But 'head' is partially structured and understood in terms of 'apple'. The attributes of the 'apple' are made salient to describe the 'head' status with the help of the metaphor ground 'coreless' (Ortony, 1979).

Verb metaphors are metaphors of action. They are formed by the contradiction produced between a verb and its logic subject or object (Su, 2000). In examples (2) and (3):

(2) It is a book to kill time for those who like it.
(3) She did not so much cook as assassinate food.

In example (2), the verb 'kill' generally means to "cause to die, put to death" in the WordNet definition and is supposed to be followed by an animate thing. Instead, it uses the inanimate thing 'time' as its object. The verb 'kill', therefore, cannot be interpreted in its literal meaning but in its metaphoric sense, that is, to loiter away. In example (3), the verb 'assassinate' usually accepts a socially prominent person as its object, but it uses the word 'food' as its companion here, indicating the unconventional action of the subject.

Examples (2) and (3) describe the English verb metaphors produced between verbs and their objects. Examples (4) and (5) show the collocations made between verbs and their subjects:

(4) "Mary saw Tom in the city."

(5) "Beijing first saw the rising of the five star red flag in Tian An Men Square."

Example (4) is a literal expression, because the person 'Mary' using the verb 'saw' as her predicate is the normal perceptual process of humankind. In contrast, 'Beijing' is a place name, which cannot perceive in a bio-physical fashion. In this way, 'Beijing' and 'saw' together generate the "semantic tension" (Jiang, 2008). This tension helps to identify example (4) as a non-literal expression. Actually, 'Beijing' also refers to the metonymic pattern place-for-people. Moreover, English verb metaphors are usually used in the newspaper headlines, for example, "China urges diplomatic efforts to resolve Iran standoff" and "Greece braces for deeper spending cuts". In these examples, our knowledge of the lexical semantics of the verbs 'kill', 'assassinate', 'see', 'urge' and 'brace', as well as our general world knowledge about them, allows us to tell the literal expressions from the non-literal ones.

Metaphors have wide applications in many NLP tasks like machine translation (Yang, 2008), question answering (Gedigian et al. 2006), information extraction (Birke, 2005) and so forth. The performance of machine translation will be improved in such cases especially if the metaphoric sentences are paraphrased instead of being translated word by word. The action models called simulation in QA systems (Narayanan and Harabagiu, 2004) can, when provided with appropriate metaphoric mapping, be extended to cover metaphoric language (Narayanan, 1997), which, in return, can improve the QA system performance. Meanwhile, many incomprehensible or irrelevant responses to queries can be avoided if the NLP system includes a method for processing non-literal language.

Metaphor identification is, however, complicated by issues including context sensitiveness, emergences of novel metaphoric forms and the need for semantic knowledge about the sentences (Krishnakumaran and Zhu, 2007). Accordingly, much research has pursued the development of computational models for metaphor identification. Fass (Fass, 1997) proposes the use of selectional preference violation technique to detect metaphors. His techniques rely on hand-coded declarative knowledge bases and a large amount of processing time. Martin (Martin, 1990) presents a knowledge-based system—MIDAS (Metaphor Interpretation, Denotation, and Acquisition System). This system based itself on conventional conceptual metaphor principles attempt to identify the source concept of the novel metaphor and map it to its target concept, generally through a series of semantic connections or an interpretive high-level representation (Birke, 2005). It is built around extensive metaphor maps and metaphor senses. In a separate body of research, Mason (2004) have grappled with the CoreMet—a corpus-based system—for discovering metaphorical mappings between concepts. CorMet is designed to identify metaphors by extracting knowledge from large, dynamically Internet corpora. Markert and Nissim (Markert and Nissim, 2002) presents a supervised classification algorithm for resolving metonymy. They see metonymy resolution as a classification task. Gedigian et al. (2006) uses a maximum entropy classifier to identify metaphors. Birke (Birke, 2005) presents the TroFi, a system for separating literal and non-literal usages of verbs through unsupervised statistical word-sense disambiguation and clustering techniques.

This paper represents a departure from the previous approaches to metaphor identification. Specifically, we describe computational metaphor identification combining a clustering approach with the statistical one. We utilize the syntactic parser[1] of Stanford University to parse the sentences in the English corpus, in order to get the subject and predicate dependency pair (nsubj), predicate and object dependency pair (dobj). We take the log likelihood measure to compute the conceptual distance between subject and predicate, predicate and object, and output the candidate non-literal sentences according to the threshold of the conceptual distance. We also use the existing metaphor resources, for example, TroFi Example Base, to compute the similarity between words, and use the k-means clustering approach to improve metaphor identification precision and recall.

This paper is organized as follows. Section 2 discusses our main techniques for identifying metaphors. Section 3 analyses the experimental results of the techniques. Section 4 presents the relevant prior work in the area of metaphor processing and identification. Section 5 discusses the observations that were made during the experiments. Finally we conclude in Section 6.

2. **Introduction. Identifying English verb metaphors using statistical and clustering approaches.** We restrict ourselves to metaphoric usages involving English verbs. In particular, we study the effect of a verb and its subject/object in a sentence. Prior to taking the two approaches, we use the Stanford University parser to analyze the dependency relations between the verb and its subject/object. We then use the statistical approach—log likelihood—to compute the conceptual distance between verb and its subject/object. In this case, the hyponym and hypernym of the subject/object are considered to describe their relation with the corresponding verbs. To improve metaphor identification efficiency, we employed the k-means clustering approach to categorize subjects/objects of training data into different clusters and then compute average similarity between words in a cluster and the words in the base metaphoric resources, that is, TroFi Example Base.

2.1. **Statistical approach.** According to the theory of selectional restrictions, the probability of the subject like 'car' typically selecting for the verb 'use' is higher than for the verb 'drink'. Hence, in the sentence "My car drinks gasoline", the affinity between 'car' and 'use' is larger than that between 'car' and 'drink'. Our hypothesis is that the larger the affinity between words is, the shorter the conceptual distance between words is. See figure 1:

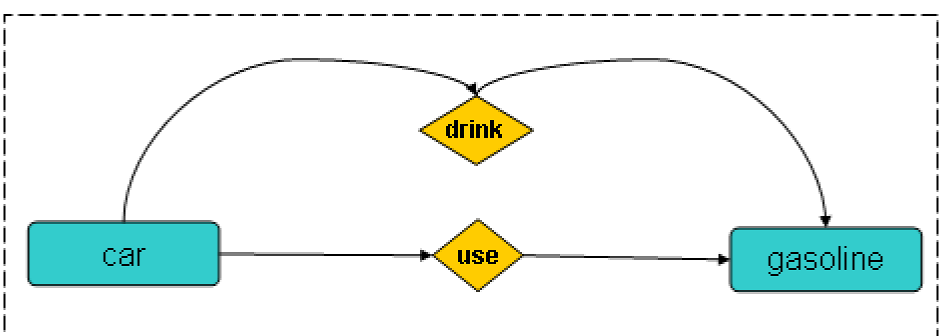

FIGURE 1. Conceptual distance between words.

In figure 1, we can see the distance between the subject 'car' and the verb 'drink' is

[1] http://nlp.stanford.edu/software/lex-parser.shtml

longer than that between ‘car’ and ‘use’. It is true to the verb ‘drink’ and the object ‘gasoline’. This lends support to our belief that the longer the distance is, the higher the probability of a metaphoric sentence is.

Based on the aforementioned belief, we then obtain the subject, object and predicate of a sentence using the Stanford University parser. We use the log likelihood measure to exploit the statistical collocation information between a verb and its subject/object. This approach is as follows. For a given pair of words X and Y (subject and verb, verb and object) and a search window W, the collocation relation of X and Y can be described by the contingency table below:

	X	!X
Y	a	C
!Y	b	D

In this table, let a be the number of the windows in which X and Y co-occur, let b be the number of windows in which only X occurs, let c be the number of windows in which only Y occurs, and let d be the number of windows in which none of them occurs, then

$$LL = 2(a\ln a + b\ln b + c\ln c + d\ln d - (a+b)\ln(a+b) - (a+c)\ln(a+c) \\ - (b+d)\ln(b+d) - (c+d)\ln(c+d) + (a+b+c+d)\ln(a+b+c+d) \quad (1)$$

The conceptual distance between X and Y is

$$\text{Distance} = 1/LL; \quad (2)$$

The equation (2) demonstrates that the less the LL is, the weaker the affinity between words is and the longer the conceptual distance is, and thus the higher the probability of a metaphoric expression is. This statistical approach, however, may be crippled by data sparseness. We handle this problem by employing the add-one smoothing method.

In this paper, we also discussed the metaphoric relationship between a verb and the hyponyms or hypernyms of the related subject or object. The idea is that if neither the subject/object nor its hyponyms or hypernyms co-occurs frequently with the verb, then the verb-subject or verb-object pair is a novel usage. Consider the following example

I guarantee you will eat your words sooner or later.

The verb ‘eat’ acts on the noun ‘word’ and makes the sentence non-literal. We find in our corpus that the objects that occur more frequently with the verb ‘eat’ are ‘food’, ‘breakfast’ and ‘lunch’, etc. Neither the noun ‘word’ nor its hyponyms/hypernyms occurs frequently with the verb ‘eat’. We, therefore, identify this verb-object dependency as metaphoric.

2.2. **Clustering approach.** Clustering is the computational task to partition given data points into clusters of equal characteristics. These clusters ideally consist of similar objects that are dissimilar to objects in other clusters (Frahling and Sohler, 2006). The most prominent and widely used clustering algorithm is Lloyd’s algorithm sometimes also referred to as the k-means algorithm. This algorithm requires each point in the d-dimensional Euclidean space be assigned to the center closest to it, and the centers are recomputed as centers of mass of their assigned points (Arthur and Vassilvitskii, 2006). This is repeated until the process stabilizes. The goal of this algorithm is to find k cluster centers and a partitioning of the points such that the sum of squared distances to the nearest center is minimized, that is, the intra-cluster distances are minimized while the inter-cluster distances are maximized (Wang, 2007; Qin, 2008). Take the verb 'eat' for example, its original object points contain 'steak', 'beef', 'lobster', 'word', 'hat', 'bowl', 'peanut', 'apple', 'seed', etc. After executing the k-means algorithm, we will get the object clusters as shown in Figure 2:

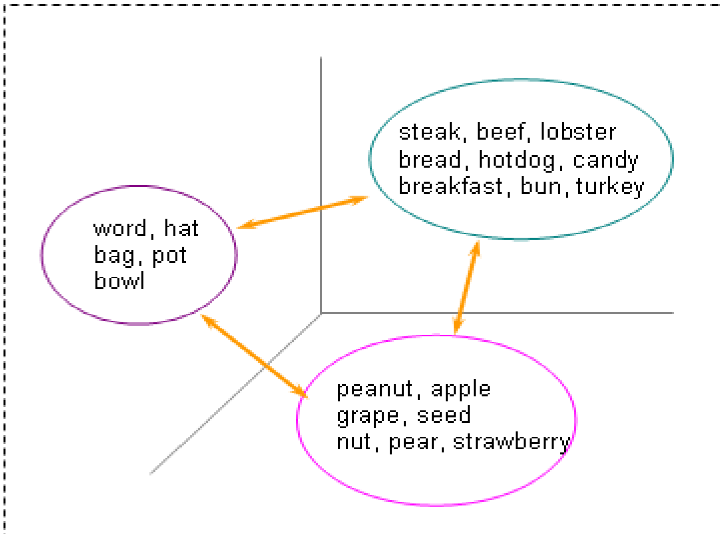

FIGURE 2. k-means clustering.

First of all, the k-means algorithm needs to solve how to choose the initial centers. Let X = $\{x_1, x_2, \ldots, x_n\}$ be a set of subject/object points in training corpus. The algorithm partitions these data points into clusters as follows:

1. By means of the WordNet resource, we use the JC similarity[2] measure to compute the similarities between every two points in X;
2. Seek the subject/object pair with the minimum similarity as the initial centers, c_i and c_j ($j \neq i$). We deem the initial centers as two clusters.
3. Set the other subject/object points in X that are closer to c_i than they are to c_j into the c_i, otherwise into c_j.
4. In each cluster, we will search for another center which is farthest to the initial center, that is, the center with the minimum similarity. The recursive process is as follows: let m be the currently chosen centers, then another center can be found by

$$c_{m+1} = \arg\max_{c_i \neq c_j} (\min_{j=1,2,3\ldots,m} [sim(c_i, c_j)]) \quad (3)$$

5. Repeat the searching process until the clusters no longer change. In this case, we set (4) as the stop condition of the iteration. Here N is the total number of the subject/object points to be clustered.

$$k < \sqrt{N} \quad (4)$$

If there are two centers equally close to a point in X, we break the tie randomly. If a cluster has no data points at the end of step 4, we eliminate the cluster and continue as

2 http://www.stanford.edu/class/cs224u/lec/224u.10.lec2.pdf

before.

2.3. **Computing average similarity.** Computing average similarity between the clusters of the training corpus and the words in the base metaphoric resources is to identify metaphoric clusters. Prior to this, we need to parse the base metaphoric resource to acquire the verb, and subject, object or both in a sentence. Take the verb ‘eat’ for example, let A and B be its two object clusters of the training corpus, and let the words ‘profit’, ‘hat’, ‘crow’, ‘word’, ‘snail’, etc. be the objects in the base metaphoric corpus. We then compute similarities between clusters of the training corpus and words in the base metaphoric resource, acquiring the average similarity, for example $Average_A$ and $Average_B$, respectively. See figure 3:

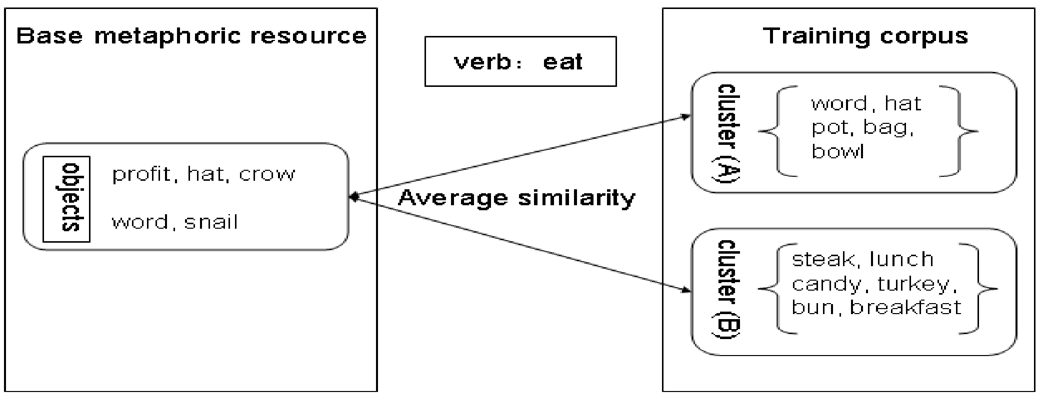

FIGURE 3. Computing average similarities.

Obviously, cluster A of the training corpus has a greater average similarity with the words in base metaphoric resource than cluster B does. Cluster A then can be recognized as non-literal.

3. **Experiment.** In this experiment, we used the training corpus of approximately 2,601,817 tokenized words (including punctuation marks) which covers the topics of transportation, tourism, sports and business. We then randomly choose from this training corpus 50 sentences for each word, including ‘destroy’, ‘eat’, ‘fill’, ‘kill’, ‘lend’, ‘miss’, ‘pass’, ‘play’ and ‘strike’. These 450 sentences are used as the testing corpus. Table 1 gives the details of the testing corpus.

TABLE 1. Testing corpus

Verb	Metaphor Testing Data		
	450	Literal	Non-literal
play	50	23	27
pass	50	39	11
eat	50	11	39
miss	50	19	31
fill	50	27	23
kill	50	23	27
lend	50	7	43
strike	50	35	15
destroy	50	23	27

Two annotators labeled the test corpus as literal (L) and non-literal (N). If two annotators see eye to eye that some sentence is literal or non-literal, then the sentence is marked as literal or non-literal. Figure 4 shows the tagged example sentences.

[N] he **struck** a false note when he arrived for the wedding in old clothes

[L] a light hammer with a rounded head that is used to **strike** percussion instruments .

[N] the news of the epidemic **struck** terror into the population

[N] The clock **strikes** the hours but not the half hours .

[L] The stone **strike** me on the side of the head .

FIGURE 4. Tagged example sentences.

In some cases, pronoun subject or object affects the performance of our algorithm. We filtered out these dependency pairs containing pronouns as their subjects or objects. In order to widen the metaphor coverage, we deleted the word index from the dependency pairs, for example, "nsubj(eats-2, Rust-1)" and "prep_into(eats-2, metal-4)" will be "nsubj(eats, Rust)" and "prep_into(eats, metal)". We also lemmatized the subject-verb and verb-object dependency pairs, for example, "nsubj(eats, rust)" and "nsubj(ate, rust)" will become "nsubj(eat, rust)" after being lemmatized. In this experiment on the nine words, our statistical algorithm has a precision of 52.49% and a recall of 63.52% with respect to literal or non-literal labels. Table 2 below shows the performance of our statistical approach for the subject-verb and verb-object dependency pairs.

TABLE 2. Performance for subject-verb and verb-object dependencies.

	Precision	Recall	F-score
play	51.85%	60.86 %	56.00 %
pass	69.44%	64.10%	66.67%
eat	24%	54.54%	33.33%
miss	61.54%	84.21%	71.11%
fill	47.06%	59.26%	52.46%
kill	42.86%	52.17%	47.06%
lend	21.74%	71.43%	33.33%
strike	63.16%	68.57%	65.75%
destroy	56.52 %	56.52%	56.52 %
Macro	52.49%	63.52%	53.58%

To validate our algorithm, we use the bi-gram measure to compute the conceptual distance between subject and verb, verb and object dependency pairs for the nine words. We then made a comparison between bi-gram measure and log likelihood measure. See figure 5:

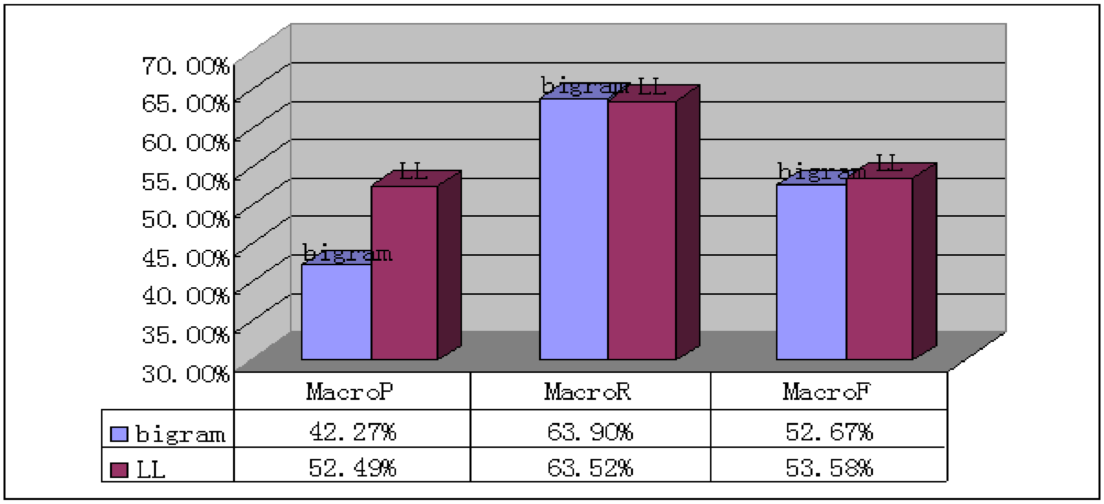

FIGURE 5. Comparison between bi-gram and LL.

Figure 5 shows that log likelihood measure presents better performance than bi-gram measure in macro-precision and macro-fallout.

In terms of clustering approach, we experimented with the TroFi Example Base (Birke, 2005). The TroFi Example Base is a collection of approximately 6783 sentences for 50 tagged English verbs. It serves both as a resource for further research and as training data for other statistical algorithms. In our experiment, we selected from TroFi Example Base these sentences that are corresponding to the nine verbs used in the statistical approach. For example,

wsj50:6427 L Mr. Hudson hears that the Chinese eat the things, and someone's looking into it.

wsj34:10862 N It just eats your soul.

In these two examples, the capital letter L and N refers to literal and non-literal respectively. We then use the Stanford parser to get the verb, subject, object or both in a sentence of this base metaphoric resource. Before we compute the average similarity, we need to cluster the subjects/objects of the sentences in the training corpus. To get the clusters whose words will form metaphoric expressions with the designated verbs, a threshold is often used, i.e. in the stage of computing average similarity, any clusters producing scores lower than a given threshold are excluded from the metaphor identification process. In this experiment, the average similarity of 1.0 is recommended as the threshold. Table 3 shows the performance of the clustering approach.

Table 3 clearly shows that some verbs (play, pass, fill, destroy) has achieved a precision or recall of over 80%. One might argue that without clustering of the subjects/objects in the training corpus, we simply need to compute the similarity between the words in the training corpus and the words in the base metaphoric resource and then get the metaphoric words in the training corpus which share the highest similarity with the words in the base metaphoric resource. To prove this, we make a comparison between clustering and non-clustering approaches in terms of macro-precision, macro-recall and macro-fallout. See figure 6:

TABLE 3. Performance of the clustering approach.

	Precision	Recall	F-score
play	85.71%	52.17 %	64.86 %
pass	67.24%	100.00%	80.41%
eat	42.86%	27.27%	33.33%
miss	62.50%	26.32%	37.03%
fill	53.19%	92.59%	67.57%
kill	60.00%	13.04%	21.43%
lend	23.52%	57.14%	33.33%
strike	73.91%	48.57%	58.62%
destroy	80.00%	34.78%	48.48 %
Macro	60.99%	50.21%	49.45 %

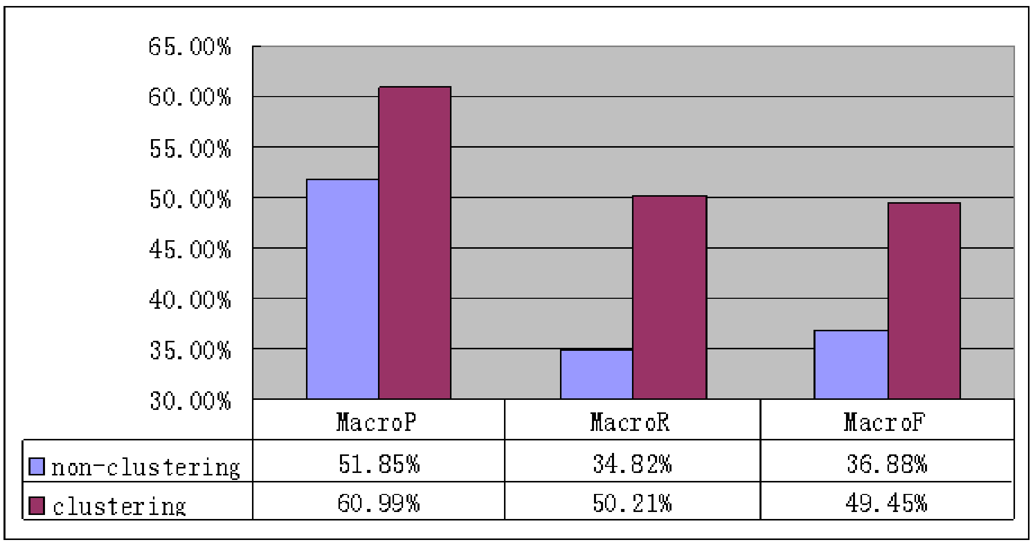

FIGURE 6. Comparison between clustering and non-clustering approaches.

Figure 6 shows that the clustering approach gains a better performance than the non-clustering one although we have taken a much stricter threshold for the clustering approach.

Finally, by comparison with the pure statistical approach, we evaluated the micro-precision, micro-recall and micro-fallout combining the statistical approach with the clustering one. See figure 7:

Figure 7 expresses that the performance of the metaphor identification was improved by incorporating the clustering approach into the statistical one.

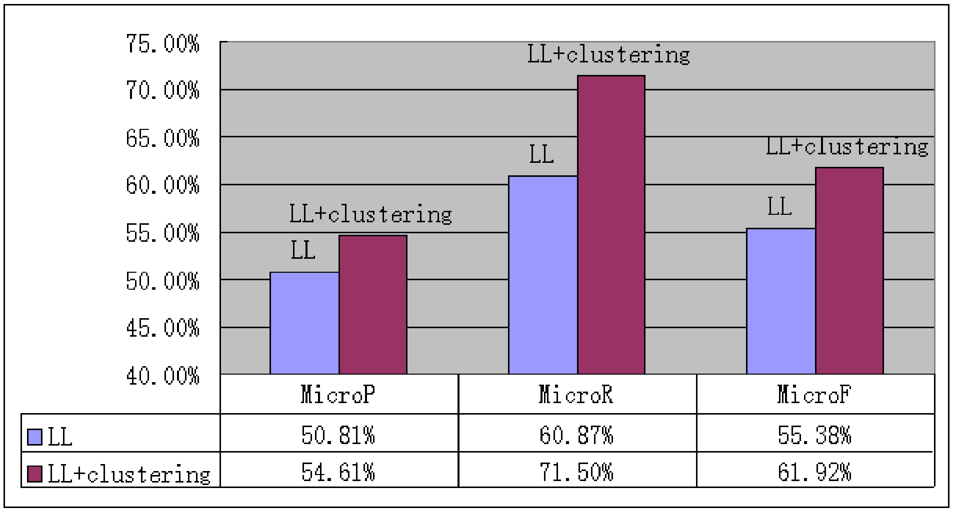

FIGURE 7. Performance of a combination of statistical and clustering approaches.

4. **Discussion.** As our experiments shows, our approaches provide a practical means of identifying metaphorical expressions. This becomes important in NLP tasks in which it can be costly and time consuming to build comprehensive semantic formula, mapping rules, metaphor maps, numerous linguistic rules, etc. In this paper, our approaches satisfy the simplicity and practicality criterion in that they do not require the mapping rules, semantic formula and the like. In particular, our approaches are capable of detecting metaphors in a sentence where the subject/object and verb are located far away from each other, i.e. long distance dependency. This, to a certain extent, reduces the noise in the sentences.

Despite the flexibility and efficiency of the statistical approach in providing metaphor candidates, there is a limit to its performance, for example, its output has to be verified by human efforts and widely used metaphoric sentences will be ignored by the statistical approach. To remedy this, we employed the clustering approach to increase the precision and recall of metaphor identification, which has been verified by our experiments. In our experiments, we also note the following issues:

(1) Parser issue

Some sentences are wrongly analyzed by the Stanford parser. For example,

She brought drinking water and tea for the passengers.

We got such dependency pair as “dobj(drinking, water)” from the parser. Normally, the dependency pair of “nn(drinking, water)” is a correct one.

(2) Prepositional phrase issue

Prepositional phrases affect the performance of our algorithm. For example,

Rust eats into metal.

After be processed by Stanford parser, we will get the following dependency pairs:

nsubj(eats-2, Rust-1)

prep_into(eats-2, metal-4)

The “prep_into(eats-2, metal-4)” dependency pair will be filtered out if we use ‘nsubj’ and ‘dobj’ pattern to get the subject-verb and verb-object dependencies. In fact, the “prep_into(eats-2, metal-4)” dependency pair might as well be a metaphoric expression.

(3) Limited coverage of the base metaphoric resource

The clustering approach relies heavily on a large manually compiled base metaphoric resource. In our experiment, the verb ‘lend’ in the base metaphoric resource simply has subjects ‘rate’, ‘polymer’, and ‘situation’. Hence, the sentence “His presence lent dignity to the occasion.” in the training corpus cannot be recognized as a metaphoric expression, because the word ‘presence’ bears a lower similarity to the words ‘rate’, ‘polymer’, and ‘situation’. To increase the metaphor identification rate, it is necessary to widen the coverage the base metaphoric resource.

5. **Related work.** The issue of computational metaphor identification has long attracted much attention from the Natural Language Processing (NLP) community (Fass, 1997; Martin, 1990; Nissim & Markert, 2002; Mason, 2004; Birke, 2005; Gedigian et al. 2006; Krishnakumaran and Zhu, 2007). A number of approaches have been suggested, including rule-based, statistical and machine-learning approaches, and have made success to various extents. Despite this research, however, computational metaphor identification still poses a tough challenge, and it has been receiving attention increasingly.

Directly related to our work is hunting elusive metaphors using lexical resources which is done by Krishnakumaran and Zhu to identify three sentence types, including ‘Subject IS-A Object’, for example, ‘He is a brave lion.’, ‘Verb acting on Noun’, for example, ‘He planted good ideas in their minds.’, and ‘Adjective acting on Noun’, for example, ‘He has a fertile imagination.’(Krishnakumaran and Zhu, 2007). Trained on the Web 1T corpus, they use the WordNet and bigram counts to automatically identify metaphors in running text. Their Type II metaphor is the same as that we have identified using the statistical approach in this paper.

We differ from each other in the approaches we are taking. We take the log likelihood measure to compute the conceptual distance between words while they take the bigram measure to calculate the collocation information between words. In our experiment, we have verified that the log likelihood measure performs better than the bigram measure in Type II metaphor identification.

Other directly related work includes the clustering approach for the unsupervised recognition of non-literal language taken by Birke (Birke, 2005). The primary task of this work is to separate literal and non-literal usages of a given target word into two distinct clusters using the word-sense disambiguation algorithm (the Karov & Edelman algorithm) and various enhancements to this basic algorithm, including similarity calculation, various learners, a voting system, SuperTag trigrams, and additional context (Birke, 2005).

We differ from each other in both our clustering algorithm and targets. We approach the recognition of non-literal language by means of the K-means algorithm, while Birke does this using the Karov & Edelman algorithm. We target at the subject and object similarity of a sentence with a great noise reduction in the sentence, but Birke targets at a sentence similarity by clustering not simply words, but also predicates, arguments, and adjuncts in the sentence where he must take great pains to reduce the sentence noises.

6. **Conclusion.** In this paper, we have reported on our experiment of identifying English verb metaphors using statistical and clustering approaches. Our approaches produced encouraging results, although further improvements are needed in terms of metaphor identification precision. Indeed, for some English verbs, high recalls above 90% have been achieved. This shows, when combined with clustering approach, the statistical method can

provide a practical and useful solution to metaphor identification. Meanwhile, further study is needed for a fuller understating of the factors affecting the performance of the statistical and clustering approaches, including the parser issue, pronoun issue, prepositional phrase issue and limited coverage of the base metaphoric resource, etc.

Acknowledgement. We would like to thank Prof. Chang and Yu, Doctor Jia for their continuous supports and everlasting encouragements. We also would like to thank our anonymous reviewers for their constructive suggestions which helped improve our paper.

REFERENCES

[1] Arthur, D. and Vassilvitskii, S. “How Slow Is the K-means Method?” Proceedings of. the 2006 Symposium on Computational Geometry (SoCG). Ed. 2006. 144–153.

[2] Birke, J. “A Clustering Approach for the Unsupervised Recognition of Nonliteral Language”. PH.D thesis, School of Computing Science, Simon Fraser University, 2005.

[3] Fass, D. “met*: A Method for Discriminating Metonymy and Metaphor by Computer.” Association for Computational Linguistics 17.1 (1991): 49-90.

[4] Frahling, G and Sohler, C. “A Fast K-Means Implementation Using Corsets*”. Proceedings of. the 2006 Symposium on Computational Geometry (SoCG). Ed. 2006. 135-143

[5] Gedigian, M. et al. “Catching Metaphors.” Proceedings of the 3rd Workshop on Scalable Natural Language Understanding. Ed. New York: 2006. 41-48.

[6] Jiang, Ling. A Contrastive Study of Metaphorical Sentences In English and Chinese. Henan: Henan University Press, 2008.

[7] Krishnakumaran S. and Xiaojin Zhu. “Hunting Elusive Metaphors Using Lexical Resources.” Proceedings of the ACL 2007 Workshop on Computational Approaches to Figurative Language. Ed. 2007. 13-20.

[8] Lakoff, G. and Mark Johnson. Metaphors We Live By. Chicago: University of Chicago Press, 1980.

[9] Mason, Zachary J. “CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System.” Computational Linguistics 30.1 (2004): 23-44.

[10] Markert K. and Malvina Nissim. “Metonymy Resolution as a Classification Task.” Proceedings of ACL-02 Conference on Empirical Methods in Natural Language Processing. Ed. 2002. 204–213.

[11] Martin, James H. A Computational Model of Metaphor Interpretation. New York: Academic Press, 1990.

[12] Ortony, A. “The Role of Similarity in Similes and Metaphors.” Metaphor and Thought. Ed. Andrew Ortony. Cambridge: Cambridge University Press, 1979. 186-201.

[13] Qin, Yonghua. Algorithm and Experiment Research of Textual Document Clustering Based on Improved K-means. New Technology of Library and Information Service 12 (2008): 73-79.

[14] Srini Narayanan and Sanda Harabagiu. “Question Answering Based on Semantic

Structures”. Proceedings of the 20th International Conference on Computational Linguistics. 2004.

[15] Srini Narayanan. “Knowledge-Based Action Representations for Metaphor and Aspect”. Ph.D. thesis, University of California at Berkeley. 1997.

[16] Su, Dingfang. Studies in Metaphor. Shanghai: Shanghai Foreign Language Education Press, 2000.

[17] Wang Hou Feng. “Clustering.” Institute of Computational Linguistics. Institute of Computational Linguistics, Peking University, 2007. Web. 21st May 2007. <http://icl.pku.edu.cn/member/wanghf/course/advanced_NLP/Lect11-Clustering.pdf>

[18] Yang, Yun. Computational Model of Chinese Metaphor Recognition and Interpretation. PH.D. thesis, Xiamen University, 2008.