Vol.2011-NL-202 No.2
2011/7/15

POS Tagging using Dependency Information

Erlyn Manguilimotan[†1] and Yuji Matsumoto[†1]

This paper presents a POS tagging approach that makes use of dependency information of a word as feature to condition a model. A part-of-speech tagger for Tagalog makes use of morphological information such as affixes and reduplication as features. However, state-of-the art sequential labeling technique cannot achieve high accuracy for Tagalog. In this work, we investigate the use of dependency head information of the words to help predict the POS tag of the word. Most existing dependency parsing assumes POS tagging as a preprocess. In this paper, we did the reverse. We apply dependency parsing without POS information, and the POS tagger tested using the output of the dependency parser. Experiments show that this approach improves the baseline scores of the POS taggers of about 1.5% for POS unigram model and 2% for POS bigram model.

1. Introduction

In natural language processing (NLP), part-of-speech tagging (POS) usually precedes syntactic analysis. POS tags of words in a sentence are used as input to parse the sentence. This approach however is subject to error propagation[1]. Other approaches on minimizing, if not totally eliminating errors, have been suggested such as integrated POS tagging and parsing described in 2) and 1). Our work does not focus on integration of POS tagging and parsing processes. Here, we investigate the use of dependency information from a dependency parser in POS tagging. We run an MST parser[3] on our data then extracted the auto-assigned heads of the words and use it as an additional feature for the POS tagger. This work is specifically done for Tagalog language; however, the approach can be adopted for other languages which shares the same characteristics of Tagalog language, specifically morphologically rich and syntactically word order free languages.

Tagalog is the national language of the Philippines[7]. As a member of the Austronesian language family, Tagalog is morphologically rich language which has affixations, stress shifting, consonant alteration, and reduplication[8]. Affixations include prefixation, infixation, suffixation and circumfixation, where two or three affixes are attached to a root word. Aside from its morphological complexity, Tagalog sentences have free word order. Predicate precedes NP complements; however, the latter take flexible positions in the sentence.

Compared to other language processing work, Tagalog NLP is relatively young. Early works on Tagalog language processing includes part-of-speech tagging [5,9], phrase-structure based sentence analysis[10] using Lexical-Functional Grammar(LFG) formalism.

2. Tagalog POS Tagging

Accuracy of POS taggers in Tagalog is much lower compared to other languages [English, 96.65%[12], Japanese 97.66%[29] and Icelandic 92.31%[28]]. A template-based Tagalog POS tagger has an accuracy of 78.3%, memory-based tagger with 77%, an HMM-based supervised tagger has an accuracy of 72%[5] and a CRF[13]-based tagger with 87.21%[9]. Problems like small sized corpora were said to attribute to the low performance of the systems. On the other hand, Tagalog's complex morphological structure also caused problems in tagging. Adjectives (e.g. *mabait* (good)) and verbs (e.g. *magalit* (to get angry at something/someone))[6] have the same affix *ma*, thus causing some ambiguities when morpho-

†1 Nara Institute of Science and Technology

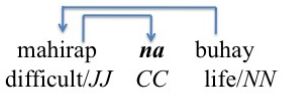

Fig. 1 Ligature *"na"*

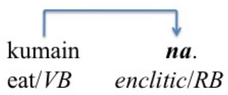

Fig. 2 Enclitic *"na"*

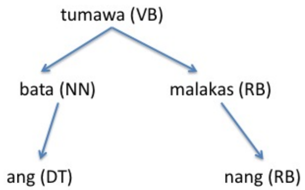

Fig. 3 Dependency tree of the Tagalog sentence *tumawa/VB ang/DT bata/NN nang/RB malakas/RB*

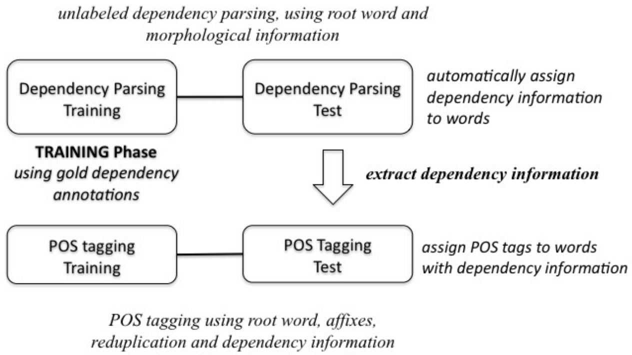

Fig. 4 Parsing - POS Tagging Process : Dependency parsing and POS tagging training uses gold annotations as dependency information. POS tag information are not included in dependency parsing training and testing.

logical information (affixes and reduplication) are use as features to condition a model in the POS tagging[9)].

Common affixes among word categories is not the only problem found in POS tagging. In Tagalog, modifiers can be an adjective or adverb depending on what is being modified by the word. Aside from modifiers, a lexeme *"na"* is found to be ambiguous[9)]. *"na"* can function as a ligature, connecting a modified with the modified word as in **Fig. 1** or simply emphasize something such as in **Fig. 2**. When POS bigram models were used, it did not help as well. POS bigram model uses the POS tag of the current word and the previous word as combined features. However, this may not be very helpful in predicting the POS tag of a word. To correctly predict a POS tag of a word, we need not only the morphological information of the previous word but also words that have syntactic relations with the word. These scenarios are suspected to have degraded the POS bigram models instead of improving the scores.

3. Dependency Parsing then POS Tagging

Figure 4 shows the general flow of our approach. We run dependency parsing using MST Parser[⋆1] by McDonald and Pereira 3) without POS tag information. Instead, features such as root words, affixes and reduplication are used. We independently run POS tagging training using the manually assigned dependency information. Features like root words, affixes and reduplication are also used. POS tagging training and testing are done using CRF-based sequential labeling tool, CRF++[⋆2].

At dependency parsing testing, the parser automatically assigns dependency heads to words in the sentence. Since MST's head information are index values corresponding to the head's position in the sentence (i.e., $0, 1, ..., n$, where 0 is the *root*), we converted these index values into coarse-grained tags (VB, NN, JJ, etc) of

⋆1 http://sourceforge.net/projects/mstparser/
⋆2 http://crfpp.sourceforge.net/

the true tags of the word when added as feature to POS tagging. For words with head 0, we assign the value *root* to know that the word is the head of the sentence. After extracting the heads from the parser, we run POS tagging testing.

4. Previous Work

Approaches to POS tagging include rule-based approach[11] and probabilistic approaches such as Hidden Markov Model (HMM), Maximum Entropy Model[12] and Conditional Random Fields (CRF)[13]. Given a sequence of words $W = w_1, \ldots w_n$ in a sentence and a set of tags T, the task is to find $t_1, \ldots, t_n$ of T that corresponds to each word in W[14].

Dependency analysis creates a link from a word to its dependents[15]. The straightforwardness of dependency analysis has been used in other NLP tasks such as word alignment[19], semantic role labeling[18]. Dependency parsers developed were either graph-based[3] or transition-based parsers[16],[17]. Syntactic analysis for different languages also use dependency parsers such as in Japanese[20],[21], English[16], Chinese[22],[23] to mention some. Usually, dependency analysis assumes preprocessing of POS tagging. However, in this work, we do dependency parsing without POS tag information and use result of dependency analysis as input to POS tagging process. The free word order nature of Tagalog language makes it fitting to use dependency parsing than phrase structure-based parsing.

5. Experiments

5.1 Experiment Data

Tagalog language resource is very limited and dependency annotated corpora was not available prior to this work. In this work, we made use of an existing POS annotated corpora from De La Salle University (DSLSU)⋆1 and annotate dependencies of the sentences. Dependency head is assigned to each token in the POS annotated corpus. The heads are chosen based on relations of word pairs such as *head-complement*, *head-modifier* or *head-specifier*. A total of 2,163 sentences, comprising of 28,825 tokens, were annotated with dependencies. The dependency annotated data are unlabeled. For POS tags, we use 85 specific tags, which is a revised version of the Rabo tag set[5]. Dependency information are coarse-grained (word category) information of the specific tags. The data is then partitioned into 5 parts for the cross validation. **Table 1** shows the data size for both training and testing.

5.2 Tagging and Parsing

Following a 5-fold cross validation process, dependency parsing is done using MST Parser and POS tagging is done using CRF++, a conditional random fields-based sequential labeling tool.

5.2.1 POS Tagging Baseline

POS tagging baseline models are trained using morphological information (root word, affixes, reduplication) of the current word w_0 and its neighboring words, w_{+1}, w_{+2}, w_{-1}, and w_{-2} as features. This set of features was chosen based on the work of Manguilimotan and Matsumoto[9]. Test results of this experiment were used as the baseline values.

5.2.2 Parsing

Using CoNLL format MST, we train our parser without using POS tag information of the word. The dependency parsing was done separately for two sets of features, *root words only*, referred to as MST1, and *root words, affixes and reduplication (if avail-*

⋆1 Center for Language Technologies of De La Salle University, Manila, Philippines

Table 1 5-fold Cross Validation Data Distribution

	Test		Train	
Parts	Tokens	Sentences	Tokens	Sentences
1	5864	419	22961	1741
2	5759	461	23066	1699
3	5827	436	22998	1724
4	5684	391	23141	1769
5	5691	456	23134	1704

able) referred to as MST2. We run the parsers with 1st order and 2nd order MST. Training phase uses manually assigned dependency annotations. This was done as reference for our parsers performance in assigning heads to words in the sentence. Parsing assumes projective dependencies only.

5.2.3 POS Tagging

POS tagging training makes use of manually assigned dependency information and is done independent from parsing. In the testing phase, we extracted auto-assigned heads of the parser in Section 5.2.2 and used as features for POS test data.

6. Results and Discussions

6.1 Parsing

Table 2 presents the MST parsing scores. Using root words with affixes and reduplication, 2nd order MST 2 obtained highest average *unlabeled attachment scores (UAS)*. Compared to dependency parsing accuracy of other languages, this score is low. One possible reason is that data size is still small. Since this work is the first dependency-based parsing for Tagalog, we do not have base scores to compare these results with against other Tagalog parsing models.

6.2 POS Tagging

After running the dependency parser we extracted the auto-assigned heads and added them as feature information to our POS tagging test data. Results of POS tagging of both baseline and with dependency information are presented in **Table 3**. The columns MST1 and MST2 in that tables refer to POS testing with dependency information extracted from parsers MST1 and MST2, respectively (refer to Section 5.2.2).

POS tagging results show that models with dependency information, whether using root words alone or with other morphological features, have higher average accuracy compared to the baseline. There is an increase of about 1.6% in unigram models and 2.0% in bigram models average scores. Secondly, dependency head information did not only improve the scores but scores of bigram models are higher compared than that of unigram models, which is not the case with the baseline. In the work of 9) bigram models also have low accuracy compared to unigram. Bigram models of MST2 (1st order and 2nd order) achieved the highest accuracy among all the models, which is consistent with the results MST results in Table 2.

6.3 Analysis and Discussion

We compare the baseline against the errors of MST 2 (2nd order). It is best to compare the baseline with models with the best score, so we chose only MST 2 (2nd order) for the comparison. Errors per word class from our baseline and MST 2 models are shown in **Fig. 5** and **Fig. 6**.

Degraded performance of bigram models in Table 3 is confirmed by the errors in Fig. 5 in verbs (VB), nouns (NN) and pronuns (PR). Adjectives(JJ) did not also show significant improvements in bigram. Generally, errors in verbs are caused by ambiguous affixes and reduplication information resulting to wrong tags such as JJ (e.g., prefix *ma*). This type of errors have been corrected by MST 2 models. However, conflicts in verbal affixes and reduplication resulted in errors within the verb class.

Table 2 Parsing results using root words only and using root words, affixes, and reduplication as features

	MST 1				MST 2			
Feature:	*root words*				*root words, affixes, and reduplication*			
Models	1st Order		2nd Order		1st Order		2nd Order	
	UAS	Complete	UAS	Complete	UAS	Complete	UAS	Complete
1	0.691	0.110	0.703	0.107	0.699	0.133	0.705	0.131
2	0.692	0.152	0.700	0.150	0.701	0.150	0.706	0.167
3	0.688	0.112	0.699	0.124	0.707	0.144	0.712	0.137
4	0.712	0.120	0.716	0.125	0.717	0.143	0.722	0.133
5	0.719	0.127	0.723	0.156	0.722	0.118	0.730	0.142
Ave	0.700	0.124	0.708	0.132	0.709	0.137	**0.715**	0.142

Table 3 POS Tagging Results for Baseline Models, MST1 and MST2 models
**U – Unigram* and *B – Bigram*
***1st – 1st Order MST* and *2nd – 2nd Order MST*

	POS Tagging									
Models	**Baseline**		**MST1**				**MST2**			
			1st		**2nd**		**1st**		**2nd**	
	U	B	U	B	U	B	U	B	U	B
1	87.44	86.85	89.12	89.08	89.20	89.15	89.05	89.23	89.23	89.34
2	86.26	86.16	87.65	87.58	87.84	87.74	87.77	87.58	87.65	87.42
3	87.21	87.11	88.77	88.81	88.82	88.71	88.79	89.18	88.75	89.08
4	87.87	87.28	88.88	88.81	89.05	88.95	89.98	88.86	88.96	88.88
5	86.91	86.62	89.19	89.33	88.89	89.36	89.17	89.38	88.98	89.49
Ave	87.13	86.80	88.72	88.72	88.76	88.78	88.71	**88.84**	88.71	**88.84**

Despite the increased average scores in MST2, POS bigram errors (Fig. 6) is almost the same as POS unigram except for adverbs (RB). The same observation is seen in MST 2 where only adverb errors decreased in bigrams. This observation would support our initial claim that bigrams do not help in improving tagger performance.

We also compare the errors between baseline unigram and MST 2 unigram as shown in **Fig. 7**. Although dependency information did not affect errors in noun (NN), pronoun (PR) and conjuncts (CC), decrease in errors are seen in verbs (VB), adjectives(JJ) and adverbs (RB). The same observation is noticed with bigrams of baseline and MST 2, with a little decrease in error in NN and PR. Noun errors are generally between personal and common nouns, which may not be resolved using head information. Number of wrong tags in pronouns did not change much with head information because pronouns can takes different heads in the data set (e.g. VB, CC, NN). These numbers should be *taken with a grain of salt* considering that some errors may have been corrected by MST2-extracted data but some correct data from baseline may have wrong tags in MST 2 models. However, the latter case is trivial compared to the number of corrected tags from the baseline.

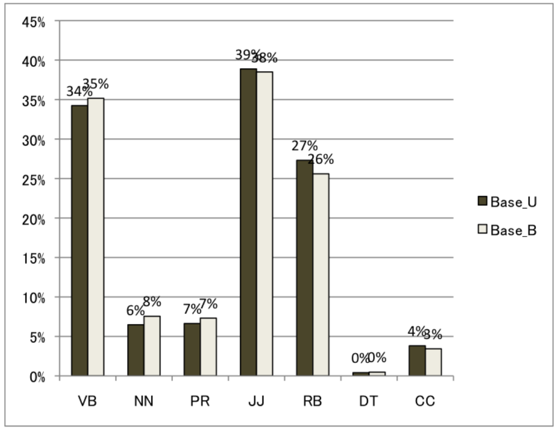

Fig. 5 Baseline Unigram and Bigram

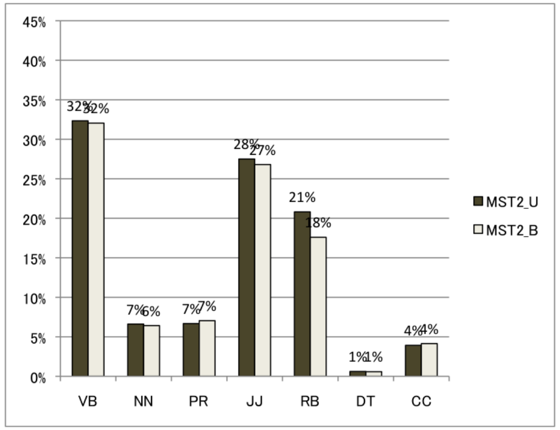

Fig. 6 MST 2 Unigram and Bigram

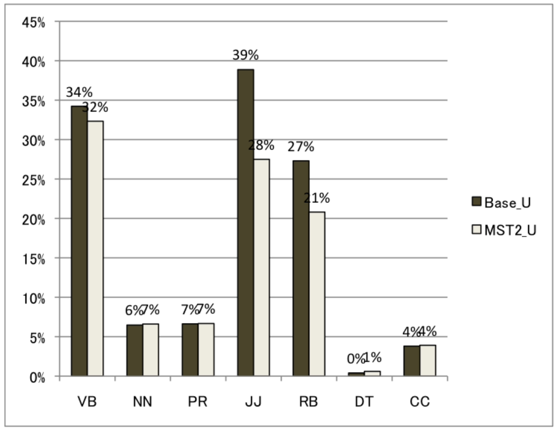

Fig. 7 Baseline Unigram and MST 2 (Unigram)

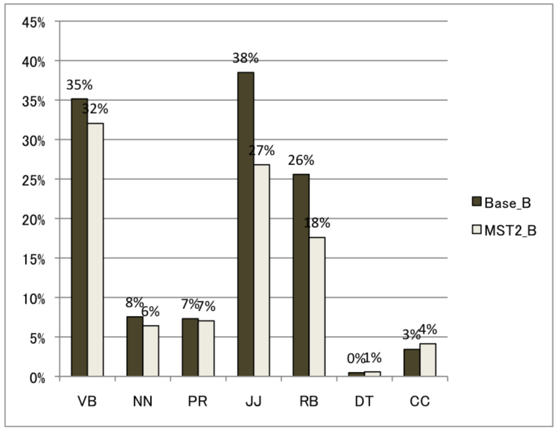

Fig. 8 Baseline Bigram and MST2 (Bigram)

Figures 7 and **8** show how dependency information improved the scores by correcting tags from verbs, adjectives and adverbs. However, this does not mean that the *corrected tags* have correct dependency heads, and correct dependency heads did not necessarily correct the tags. The case is, the MST parser may have assigned arbitrary head to a word, however, in the grammatical sense, that assigned head is still a *possible head* of the word.

Although bigram models in the results obtained better scores compared to unigrams, we cannot directly imply that the adjacent words have been the sole factor to this increase. Instead, we can liberally imply that dependency heads assigned to words or *heads*, true heads or not, has affected our POS tagging results. Words with ambiguous tags such as JJ and RB have shown improvements when head information are used.

7. Summary

In this work, we tried a new approach in improving POS tagging scores. We make use of dependency information of the word as an addition feature to condition POS tagging model. The baseline POS tagging system uses root words, affixes and reduplication, if available, as features. We run MST parser on our data without using POS tag information then extracted dependency information from the parsing results and append this information as a feature to the POS tagging.

Our results show that dependency head information, while they may not always be true, helped in predicting the POS tag of a word. Increase of 1.6% for unigram and 2.0% for bigram were achieved. Bigram models also showed higher scores compared to unigram, which is not the case in the baseline models. Whether the parser assigned a correct head of the word or not, some ambiguous POS tags were corrected by these heads. Words that has ambiguous tags such as adjectives and adverbs, as well as words with common affixes (e.g., verbs vs. adjectives) showed increase in scores using dependency information.

References

1) Cao, H, Zhang, Y., and Isahara H.: Study on Architectures for Chinese POS Tagging and Parsing, *Advances in Natural Language Processing and Applications Research in Computing Science* 33, pp. 55-66 (2006).
2) Rush, A. et al.: On Dual Decomposition and Linear Programming Relaxations For Natural Language Processing, *In the Proceedings o the 2010 Conference on Empirical Methods in Natural Language Processing* (2010).
3) McDonald,R. and Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms, *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)* (2006).
4) Ryan McDonald, Koby Crammer, and Fernando Pereira. McDonald,R. Crammer, K., and Pereira, F.: Online Large-Margin Training of Dependency Parsers, *In the Proceedings of the 43rd Annual Meeting o the Association for Computational Linguistics* (2005).
5) Miguel, D. and Roxas, R.E.: Comparative Evaluation of Tagalog Part-of-Speech Taggers, *In the Proceedings of the Proceedings of the 4th National Natural Language Processing* (2007).
6) Cheng, C. and Rabo, V.: TPOST: A Template-Based, n-gram Part-of-Speech Tagger for Tagalog, *Journal of Research in Science, Computing and Engineering (JRSCE)*, Vol. 3, No. 1. DLSU-M (2006).
7) Bonus, D.E.: The Tagalog Stemming Algorithms (TagSA), *In the Proceedings of the Natural Language Processing Research Symposium*, DLSU, Manila (2003).
8) Nelson, H.: A Two-Level Engine for Tagalog Morphology and a Structured XML Output for PC-KIMMO (online), available from ¡ http://contentdm.lib.byu.edu/ETD/image/etd465.pdf¿
9) Manguilimotan, E. and Matsumoto, Y.: Factors Affecting POS Tagging for Tagalog, *In the Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pp. 803-810 (2009).

10) Borra, A., et al.,: FiSSAn: Filipino Sentence Syntax and Semantic Analyzer, *In the Proceedings of the 7th Philippine Computing Science Congress, PCSC* (2007).
11) Eric Brill: Transformation-based Error-Driven Learning and Natural Language Processing: a Case Study in Part-of-Speech Tagging, *Computational Linguistics*, volume 21. MIT Press, Cambridge, MA, USA. pp. 543-565. (1995)
12) Adwait Ratnaparkhi: A Maximum Entropy Model for Part-of-Speech Tagging, *In Proceedings of the Empirical Methods in Natural Language Processing*, pp. 133-142 (1995).
13) Lafferty, J. McCallum, and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *In Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282-289 (2001).
14) Roark, B. and Sproat, R.: *Computational Approaches to Morphology and Syntax*, Oxford University Press, New York, USA (2007).
15) Covington, M.: A Fundamental Algorithm for Dependency Parsing, *In Proceedings of the 39th Annual ACM Southeast Conference*, pp. 95-102 (2001).
16) Nivre, J. and Scholz, M.: Deterministic Dependency Parsing of English Text, *In Proceedings of the 20th International Conference on Compurational Linguistics* (2004).
17) Yamada, H. and Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines, *In Proceedings of 8th International Workshop on Parsing Technologies*, pp. 195-206 (2003).
18) Kadri Hacioglu: Semantic Role Labeling Using Dependency Trees, *In Proceedings of the 20th International Conference on Compurational Linguistics* (2004).
19) Ma, Y. et al.,: Improving Word Alignment Using Syntactic Dependencies, *In Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pp. 69-77 (2008).
20) Kudo, K. and Matsumoto, Y.: Japanese Dependency Analysis using Cascaded Chunking, *In Proceedings of the Conference on Computational Natural Language Learning* CoNLL (2002).
21) Iwatate, M., et al., : Japanese Dependency Parsing Using a Tournament Model, *In Proceedings of the 22nd International Conference on Computational Linguistics* (Coling 2008), pp. 361-368 (2008).
22) Chen, W., et al., : Using Short Dependency Relations from Auto-Parsed Data for Chinese Dependency Parsing, *ACM Transactions on Asian Language Information Processing (TALIP)*, volume 8 issue 3 (2009).
23) Yu, K., et al., Chinese Dependency Parsing with Large Scale Automatically Constructed Case Structures, *In the Proceedings of the 22nd International Conference on Computational Linguistics* (Coling 2008), pp. 1049-1056 (2008).
24) Paul Kroeger: Phrase Structure and Grammatical Relations in Tagalog Center for the Study of Language and Information, Stanford California.
25) Schachter, P. and Otanes, F.: *Tagalog Reference Grammar*,University of California Press (1972).
26) Mistica, M. and Baldwin, T.: Recognising the Predicate-Argument Structure of Tagalog, *In the Proceedings of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies* (NAACL HLT), Short Paper (2009).
27) McDonald, R. and Nivre, J.: Characterizing the Errors for Data-Driven Dependency Parsing Models, *In the Proceedings of the Empirical Methods in Natural Language Processing and Natural Language Learning* EMNLP-CoNLL) (2007).
28) Loftsson, H., et al.,: Improving the POS Tagging Accuracy of Icelandic Text, *In the Proceedings of the 17th Nordic Conference of Computational Linguistics* (NODALIDA-2009).
29) Kudo, T., et al., : Applying Conditional Random Fields to Japanese Morphological Analysis, *In the Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, 230237 (2004).