POS Tagging of English Particles for Machine Translation

Ma Jianjun
School of Computer Science and Technology and School of Foreign Languages
Dalian University of Technology
Dalian, China, 116024
majian@dlut.edu.cn

Huang Degen
School of Computer Science and Technology
Dalian University of Technology
Dalian, China, 116024
huangdg@dlut.edu.cn

Liu Haixia
School of Computer Science and Technology
Dalian University of Technology
Dalian, China, 116024
liuhaixorama@foxmail.com

Sheng Wenfeng
School of Foreign Languages
Dalian University of Technology
Dalian, China, 116024
shengwenfeng123@126.com

Abstract

Part-of-speech tagging is a crucial preprocessing step for machine translation. Ambiguity in the natural language processing has made POS tagging hard. And particles are a major cause of ambiguity. But current studies have limited particles in narrow sense. Therefore, this study presents an English POS tagger basically addressing the tagging of particles in broad sense. A definition of particles in broad sense is given, a small size of 998k English annotated corpus in business domain is built, the maximum entropy model is adopted and rule-based approach is used in post-processing. Experiments show that our tagger achieves an F-score of 90.87% in closed test and 87.24% in open test, which is a quite satisfactory result.

1 Introduction

Part-of-speech (POS) tagging is the process in which a proper part of speech is assigned to each word for a sequence of words. The task of POS tagging is very important for various text understanding applications including machine translation, question answering and Internet search. For machine translation, the accuracy of POS tagging is a crucial preprocessing step for a high-quality translation. However, ambiguity in the natural language processing makes POS tagging hard. For example, it's often difficult to distinguish particles from prepositions or adverbs (Santorini, 1990).

Particles refer to those prepositions or adverbs such as *in*, *on*, *up*, or *down* when they combine with verbs to form phrasal verbs (Sinclair, 2000). Errors often occur in machine translation when a particle is recognized as a preposition as in *They might at any time turn against their masters*, where *against* is a particle, and *turn against* forms a phrasal verb meaning distrust, which should be understood and translated as a whole. If *against* is recognized as a preposition, then *against their masters* is very likely to be considered as NP in further parsing and translating, and *turn* as the main verb individually, thus causing misunderstanding. This is exactly the case when this English sentence is translated into Chinese by GOOGLE online machine translation system[1]. Examining its Chinese output *他们可能随时打开对他们的主人*, it won't be hard to find the cause of ambiguity is *against*, which is translated into *对* as a preposition. And in turn, *turn* is misunderstood and translated individually as *打开*. Since particles form part of the main verb of a sentence, this ambiguity will cause more serious problems than other cases. So it's worthwhile to improve the POS tagging of particles for the benefit of machine translation.

In POS tagging research, particles are tagged RP as in the pioneering Brown Corpus (Greene and Rubin, 1971) and Penn Treebank (Marcus et al. 1993), or AVP as in CLAWS (Garside and Smith, 1997). In current studies on verb particle constructions, they either use sophisticated parsers, includ-

[1] Available at http://translate.google.cn/#.

ing tagger based method, to perform extraction from corpora (Baldwin and Villavicencio, 2002; Kim and Baldwin, 2006), or use the web as the corpus (Villavicencio, 2003; Kummerfeld and Curran, 2008). But their definitions of particles have obvious limitations. On most occasions, particles are defined in narrow sense, that is, they are limited to a preposition or adverb when only one or two participants are involved in the process. Particles mainly refer to those in intransitive verb-particle construction and transitive verb-particle construction, as is stated by Baldwin and Villavicencio (2002). For example, *Income tax is coming down.* In this sentence, only one participant (*income tax*) is involved in the process of *come down*, and *down* is recognized as a particle. Another case in point is in *She ran her best friends down*, where two participants of *she* and *her best friends* are involved in the process of *run down*. When three or more participants are involved in one process, these taggers fail to distinguish the particle from the preposition. For instance, *He informed Barbara of his objections.* In this case, three participants are involved in the process of *inform*: *he*, *Barbara*, *his objections*. And *of* serves as a particle, which occurs in collocation with the verb *inform*, used to connect two participants. Disambiguation errors occur again if *of* is considered as a preposition. From the GOOGLE Chinese output of this English sentence 他告诉他的反对芭芭拉, there is no doubt that the cause of error is *of*, which is understood as a preposition in NP *Barbara of his objections*, translated into 他的反对芭芭拉. Actually, this error is fatal, for the translation like this makes no sense in terms of the communicative purpose.

This paper, therefore, presents an English POS tagger basically addressing the tagging of particles in broad sense. The definition of particles in broad sense is given in Section 2. As to the POS tagging method, many rule-based, statistical and machine learning methods have been applied currently, such as transformation-based error-driven learning (Brill, 1995), transformation-based learning (Bach et al., 2008), neural networks (Zamora-Martinez et al., 2009), decision trees (Schmid, 1994; Wang, 2010), entropy guided transformation learning (ETL) (dos Santos et al., 2008), memory-based learning (Daelemans, 1996), maximum entropy models (Ratnaparkhi, 1994; Huang, 2009), hidden Markov models (HMM) (Brants, 2000; Collins, 2002), HMM with rule based approach (Zin, 2009), the Markov family models (Yuan, 2010), and latent analogy (Bellegarda, 2010). Considering the small size of our corpus, the maximum entropy model is adopted and rule-based approach is used in post-processing, the details of which are presented in Section 3. Section 4 reports the results of experiments and some discussions. Finally, some conclusions are given in Section 5.

2 Particles in broad sense

The particles we target in this study are particles in broad sense. We define a particle as a preposition or a directional adverb when it combines with the verb to form a phrasal verb. For the purpose of machine translation, unlike the Penn Treebank (Santorini, 1990), we adopt the idiomaticity of a collocation as a criterion that a word is a particle. That is, when a preposition or a directional adverb is specially required by a previous verb, and occurs in collocation with the verb, it is defined as a particle.

Therefore, a particle may refer to a preposition on two surface structures:

Structure 1: "V prep n"

e.g. *They might at any time turn against*/RP *their masters.*

Structure 2: "V n prep n"

e.g. *He informed Barbara of*/RP *his objections.*

It may also include an adverb on another two surface structures:

Structure 3: "V adv"

e.g. *Income tax is coming down*/RP.

Structure 4: "V n adv"

e.g. *She ran her best friends down*/RP.

Similarly, in terms of participants of a process, a particle may not only occur in an one-participant process (e.g. *Why don't you come by*/RP*?*), and a two-participant process, either in a joint configuration (e.g. *They might at any time turn against*/RP *their masters.*) or in a split configuration (e.g. *She ran her best friends down*/RP.), but also occur in a three-participant process (e.g. *I put her suitcase on*/RP *the table.*).

Obviously, according to this definition, the verb and particle can be contiguous, as in *Income tax is coming down*/RP, and non-contiguous, as in *I put her suitcase on*/RP *the table.* The second aspect makes it more difficult to distinguish a particle from a preposition or an adverb.

These characteristics also show that particles in broad sense vary with verbs and the context should be taken into consideration in the process of tagging. And in actual texts, the sentences are supposed to be longer and more complicated, which adds more difficulty to the tagging of English particles.

In our tagger, particles are tagged RP, and the prepositions on other occasions are tagged INP and adverbs RB.

3 Methods

In our study, a small size of 998k English annotated corpus in business domain is built, the corpus is pre-processed using one Stanford Tagger, the maximum entropy model is adopted and rule-based approach is used in post-processing.

3.1 Corpus

This study is considered as a preprocessing step for an English-Chinese machine translation project in the domain of business, but no large, manually annotated bilingual corpus is available for training, so a small corpus of 998k is built, which consists of 10059 sentences. Those sentences come from two sources: 9 publications in business field and 7 internet websites, covering 14 specific situations in business, such as inquiry and reply, offer, counter-offer, order, contract, packing, shipping, payment, claim, insurance, transport, agency, establishing business and marketing.

The corpus is manually tagged according to the Penn Treebank tag set (Marcus et al, 1993) for training and testing. Two changes are made in the Penn Treebank tag set. One change is in the distinction between preposition and subordinating conjunction; IN is further distinguished into INP (Preposition) and INC (Subordinating conjunction). The other change is about the word *to*. TO just refers to the infinitive in our tagger. When it is used as a preposition or a particle, it is tagged INP or RP respectively.

Table 1 shows the detailed information of our corpus, with the total token being 198053, and tokens of RP 5197, which is similar to the tokens of RB (6727). Since a particle itself is either a preposition or an adverb according to its definition, so this tagging job is mainly to distinguish the 5197 particles from the rest 25677 prepositions and adverbs.

	Tokens
Overall	198053
INP	18950
RB	6727
RP	5197

Table 1. Corpus Information

3.2 Pre-processing using a Stanford POS tagger

In order to solve the problem of data sparseness, the corpus is pre-processed by using one Stanford POS tagger to get the tags as one important feature for the Maximum Entropy model. The Stanford tags are chosen as one feature for two reasons. One reason is that the tagger is trained on a large corpus, WSJ sections 0-18, with a best result of 97.18% correct. The other is because of its tag set. It also adopts Penn tag set, which is used in our tagger, with two major changes being made. Therefore, choosing the Stanford tags as one feature can not only help improve the efficiency of machine learning, but reduce data sparseness, which is caused by the small size of our training corpus.

Definition	Tagger	Tag set	Tokens of RP
particle in narrow sense	Stanford tagger	Penn tag set	285
particle in broad sense	our tagger	our tag set	5197

Table 2. Particles in two senses

We choose one Stanford English POS tagger, bidirectional-wsj-0-18.tagger, which achieves the best performance among the three English taggers. This tagger is trained on WSJ sections 0-18 using bidirectional architecture and including word shape features, with a performance of 97.18% correct on WSJ 19-21[2]. We use this tagger to tag on our training corpus. Table 2 shows that this tagger finds only 285 particles as opposed to 5197 particles to be tagged in our tagger, which further illustrates the definition of particle in broad sense, with particles in narrow sense accounting for only 5.5% of particles in broad sense based on our corpus.

[2] Available at http://nlp.stanford.edu/software/tagger.shtml

3.3 The Baseline Maximum Entropy Model

The Maximum Entropy Model is adopted in this study. The principle of Maximum Entropy is first proposed by Jaynes (1957) which states the correct distribution p(a, b) is that maximizes entropy or uncertainty, subject to the constraints. A conditional Maximum Entropy model, also known as a log-linear model, has the following form:

$$p_\lambda(y \mid x) = \frac{1}{Z_\lambda(x)} \exp\left[\sum_i \lambda_i f_i(x, y)\right] \quad (1)$$

$$Z_\lambda(x) = \sum_y \exp\left[\sum_i \lambda_i f_i(x, y)\right] \quad (2)$$

Where the functions f_i are the features of the model, usually a binary-valued function. λ_i is the weight of f_i or the parameters of the model, Z(x) is a normalization constant. This formula can be derived by choosing the model with maximum entropy from a set of models that satisfy a certain set of constraints. Given k features, the constraints which represent that the model's feature expectation is equal to the observed feature expectation, have the following equation:

$$E_p f_i = E_{\hat{p}} f_i \quad (3)$$

Given the constraints, the parameter estimation of the Maximum Entropy model becomes an optimization problem. The parameters can be obtained via an algorithm called Generalized Iterative Scaling (Darroch and Ratcliff, 1972).

No.	Condition	Features
1	General	w_i＝X & t_i＝T
2	General	st_i＝X & t_i＝T
3	General	st_{i-1} st_i＝XY & t_i＝T
4	General	st_i st_{i+1}＝XY & t_i＝T
5	General	w_{i-1}＝X & t_i＝T
6	General	w_{i+1}＝X & t_i＝T

Table 3. Feature template of ME model

Three features are used in this model: word, Stanford tag, and our tag. The features that define the constraints on the model are obtained by instantiation of feature templates, as shown in Table 3, where w_i, st_i, t_i are used to denote the word, Stanford tag, and our tag respectively, w_{i-1} and w_{i+1} denote the words just before and after the token respectively, and similarly, st_{i-1} and st_{i+1} the Stanford tags of the words w_{i-1} and w_{i+1} respectively.

3.4 Post-processing with rules

After going through the output, we find errors still occur on the following 3 occasions. The first is when a particle immediately follows a particular verb, such as *coincide with*, *complain of*, or *consist of*. In this case, these particles are tagged INP instead of RP. The second is when a particle is far away from the verb, such as *inform ... of ..., reduce ... by ..., extend ... to ..., advise ... of ...,* and *assure ... of ....* On this occasion, these particles form a strong collocation with the verbs, but are tagged INP instead. The last is when some prepositional phrases started with prepositions of *for, to, with* are used as adjuncts in the sentences, such as *for your reference*, *to the contrary*, and *with a view to*. These prepositions should be tagged INP instead of RP.

Therefore, three collocation banks are manually created accordingly: VB+RP bank, VB+NN+RP bank, and INP+NN adjunct bank. The following rules are adopted in post-processing.

Rule 1: When a preposition or two or an adverb immediately follows a verb, search the VB+RP collocation bank. If these two or three words as a whole match one collocation in the bank, then the preposition(s) or adverb is tagged RP. All the verb forms are included in the search.

Rule 2: When a verb that matches a verb in VB+NN+RP bank occurs, search the words to the right of it to find the first of an expected word described in the bank till another verb appears or a that clause appears. If the word does occur, then it is tagged RP. All the verb forms are included in the search.

Rule 3: When a group of words match exactly a collocation in INP+NN adjunct bank, then the corresponding preposition is tagged INP.

Among the three rules, Rule 3 takes the highest priority in processing, with Rule 1 being the next and Rule 2 the last.

4 Results and discussion

In order to test the system performance, both closed test and open test are made on our corpus.

4.1 Closed test

A closed test is made on our corpus, and Table 4 compares the results of the baseline ME model and those after post-processing. Table 4 shows that the closed test achieves a precision of 94.12%, a recall of 87.84% and an F-score of 90.87%. As is shown in Table 4, the rule-based approach has especially increased the recall by 3.04%, while the precision is increased by 1.38% and the F-score 2.28%.

	Precision (%)	Recall (%)	F-Score (%)
ME	92.74	84.80	88.59
ME + RuleBased	94.12	87.84	90.87

Table 4. Close test results

After analyzing the error reports, we find that errors occur most frequently with three confusing words: *to, with* and *for*. The reason is that these three words can be used as particles and co-occur with verbs, can be used in a prepositional phrase which functions as a post-modifier, and in a prepositional phrase which serves as an adjunct. For example:

*If you could not supply the goods enquired for, would you please refer our enquiry to/*RP *the interested parties for/*INP *attention.*

In this sentence, *to* is a particle, which co-occurs with the verb *refer*, while *for* is a preposition, and *for attention* forms an adjunct. Therefore, it's hard to distinguish RP from INP. Though the rule-based approach helps improve the system performance, problems remain unsolved when the collocation banks are not complete.

Another problem lies in the distance between the particle and the corresponding verb. Again when the particle is far away from the verb, it's very hard to distinguish it from a preposition. Though a VB+NN+RP bank is built and a rule is adopted in post-processing, it's still likely that a particle seems to form a stronger collocation with the noun just before it, and so it is tagged INP more often than RP. For instance:

Would you please inform us in detail of/RP *its price, terms of payment and terms of shipment?*

In this example, *in detail* as an adjunct is embedded in the VB+NN+RP structure: *inform us of*, which adds difficulty to tagging. In the output, *of*, which is supposed to be tagged RP, seems to have a stronger collocation with the noun *detail*, and then is tagged INP, with *of its price* being considered as a post-modifier of *detail*. Considering this possibility, we did a very careful job when we established the VB+NN+RP bank. Patterns were selected only when the particle and the verb form very strong collocation, which, on the other hand, limits the application of Rule 2 mentioned in Section 3.4.

4.2 Open test

Five cross tests are made in open test in order to have a reliable result. The average score is chosen as the final result. The corpus is divided into 5 groups, with each group equally covering all the 14 situations. In each open test, one group is chosen as the testing corpus and the rest four groups are the training corpus. Table 5 gives the details of the training and testing corpora of each test. Table 6 presents the average results, with the final F-score being 87.24%, precision 90.93% and recall 83.86%. According to Table 6, like the close test, the rule-based approach has increased F-score by 2.6% in open test, and precision and recall are increased by 1.65% and 3.39% respectively.

	Testing corpus	Training corpus
Test 1	Group 1	Groups 2, 3, 4, 5
Test 2	Group 2	Groups 1, 3, 4, 5
Test 3	Group 3	Groups 1, 2, 4, 5
Test 4	Group 4	Groups 1, 2, 3, 5
Test 5	Group 5	Groups 1, 2, 3, 4

Table 5. Training and testing corpora

	Precision (%)	Recall (%)	F-Score (%)
ME	89.28	80.47	84.64
ME + RuleBased	90.93	83.86	87.24

Table 6. Open test final results

Figure 1 further compares the performance of each test, including both before and after the post-processing. It's obvious that the results very slightly with the change of training and testing corpora. Sometimes when the precision is high, the recall may be low, as in Test 5, thus the F-score being most reliable.

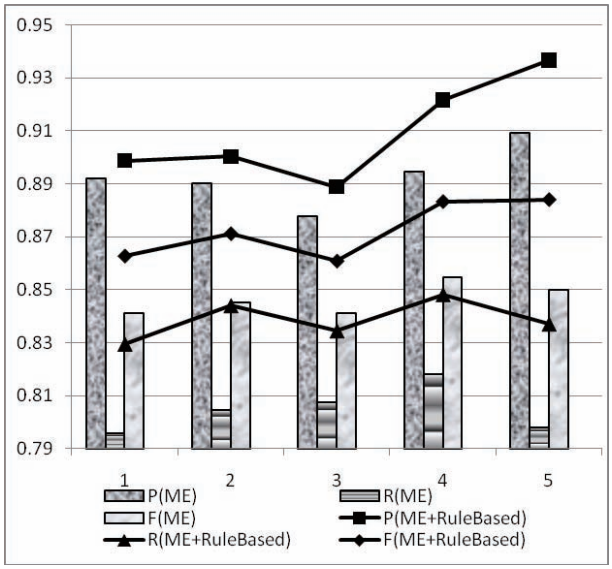

Figure 1. Results of each open test

5 Conclusion

This study presents an English POS tagger basically addressing the tagging of particles in broad sense. A new definition is clearly given. A small size of 998k English annotated corpus in business domain is built, and pre-processed by using one Stanford POS tagger to get the tags as one important feature. The maximum entropy model is adopted and rule-based approach is used in post-processing. Both closed test and open test are made on the corpus. Experiments show that our tagger achieves an F-score of 90.87% in closed test and 87.24% in open test, which is a quite satisfactory result. The rule-based approach increases the F-score by 2.28% and 2.6% respectively.

This study is worthwhile for English-Chinese machine translation, particularly noun phrase recognition, simply because a particle is not part of a noun phrase, but part of a verb phrase. Of course, the system performance can be further improved by enlarging the corpus and applying more proper rules in post-processing, which is the focus of our future study.

Acknowledgments

This study is supported by "the Fundamental Research Funds for the Central Universities (DUT10RW202)". We gratefully acknowledge this support. We are also grateful to Li Zezhong for his programming support.

References

Bach, Ngo Xuan, Le Anh Cuong, Nguyen Viet Ha, and Nguyen Ngoc Binh. 2008. Transformation Rule Learning without Rule Templates: A Case Study in Part of Speech Tagging. In *Proceedings of 7th International Conference on Advanced Language Processing and Web (ALPIT 2008)*, pages 9-14, Dalian.

Baldwin, Timothy and Aline Villavicencio. 2002. Extracting the Unextractable: A Case Study on Verb-particles. In *Proceedings of the 2002 Conference on Natural Language Learning*, pages 1–7, Taipei.

Bellegarda, Jerome R. 2010. Part-of-Speech Tagging by Latent Analogy. *IEEE Journal of Selected Topics in Signal Processing*, 4(6):985-993.

Brants, Thorsten. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP2000)*, pages 224-231, Seattle, WA.

Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4): 543–564.

Collins, Michael. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Philadelphia, PA.

Daelemans, Walter, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A Memory-Based Part of Speech Tagger-Generator. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14–27, Copenhagen, Denmark.

Darroch, John N. and Douglas Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470-1480.

dos Santos, Cicero Nogueira, Ruy L. Milidiu, and Raul P. Renteria. 2008. Portuguese Part-of-Speech Tagging Using Entropy Guided Transformation. In *Proceedings of 8th International Conference on Computational Processing of the Portuguese Language*, pages 143-152, Aveiro.

Garside, Roger and Nicholas Smith. 1997. A hybrid grammatical tagger: CLAWS4. In Roger Garside, Geoffrey Leech, and Anthony McEnery, editors, *Corpus Annotation: Linguistic Information from Computer Text Corpora*. Longman, London, pages 102-121.

Greene, Barbara B. and Gerald M. Rubin. 1971. *Automatic Grammatical Tagging of English.* Technical Report, Department of Linguistics, Brown University.

Huang, Heyan and Zhang Xiaofei. 2009. Part-of-Speech Tagger Based on Maximum Entropy Model. In *Pro-*

ceedings of 2nd IEEE International Conference on Computer Science and Information, pages 26-29, Beijing.

Jaynes, Edwin T. 1957. Information Theory and Statistical Mechanics. *Phys. Rev.*, 106(4): 620-630.

Kim, Su Nam and Timothy Baldwin. 2006. Automatic Identification of English Verb Particle Constructions Using Linguistic Features. In *Proceedings of the 2006 Meeting of the Association for Computational Linguistics: Workshop on Prepositions*, pages 65–72, Trento.

Kummerfeld, Jonathan K. and James R. Curran. 2008. Classification of Verb-Particle Constructions with the GoogleWeb1T Corpus. In *Proceedings of the Australasian Language Technology Association Workshop (ALTA 2008)*, pages 55-63, Tasmania.

Marcus, Mitchell P., Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ratnaparkhi, Adwait. 1996. A Maximum Entropy Model for Part-of Speech Tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference(EMNLP-96)*, pages 133-142, Philadelphia, PA.

Santorini, Beatrice. 1990. *Part-of-speech tagging guidelines for the Penn Treebank Project.* Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

Schmid, Helmut. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Sinclair, John. 2000. *Collins COBUILD Grammar Patterns 2: Verbs.* Shanghai Foreign Language Press, Shanghai, page xxi.

Villavicencio, Aline. 2003. Verb-particle Constructions and Lexical Resources. In *Proceedings of the Meeting of the Association for Computational Linguistics: 2003 workshop on Multiword expressions*, pages 57–64, Sapporo.

Wang, Yongsheng. 2010. Research on part-of-speech tagging using decision trees in English-Chinese machine translation system. *Computer Engineering and Applications*, 46(20):99-102.

Yuan, Lichi. 2010. Improvement for the automatic part-of-speech tagging based on hidden Markov model. In *Proceedings of 2nd International Conference on Signal Processing Systems (ICSPS 2010)*, pages 744-747, Dalian.

Zamora-Martinez, Francisco, Maria J. Castro-Bleda, Salvador Espana-Boquera, Salvador Tortajada and P. Aibar. 2009. A Connectionist Approach to Part-of-Speech Tagging. In *Proceedings of 1st International Joint Conference on Computational Intelligence (IJCCI 2009)*, pages 421-426, Funchal Madeira.

Zin, Khine. 2009. Hidden Markov Model with Rule Based Approach for Part of Speech Tagging of Myanmar Language. In *Proceedings of 3rd International Conference on Communications and Information*, pages 123-128, Vouliagmeni.