The Role of PP Attachment in Preposition Generation

John Lee[1] and Ola Knutsson[2]

[1] Spoken Language Systems
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, USA
jsylee@csail.mit.edu
[2] School of Computer Science and Communication
Royal Institute of Technology, Stockholm, Sweden
knutsson@csc.kth.se

Abstract. This paper is concerned with the task of preposition generation in the context of a grammar checker. Relevant features for this task can range from lexical features, such as words and their part-of-speech tags in the vicinity of the preposition, to syntactic features that take into account the attachment site of the prepositional phrase (PP), as well as its argument/adjunct distinction. We compare the performance of these different kinds of features in a memory-based learning framework. Experiments show that using PP attachment information can improve preposition generation accuracy on Wall Street Journal texts.

1 Introduction

Preposition usage is among the more frequent types of errors made by non-native speakers of English. In an analysis of texts [1], written by students in English-as-a-Second-Language classes, errors involving prepositions form the largest category, at about 29%[1]. A system that can automatically detect and correct preposition usage would be of much practical and educational value. Research efforts towards building such a grammar checking system have been described in [2], [3], and [4].

When dealing with preposition errors, the system typically makes two decisions. First, a *preposition generation* model needs to determine the best preposition to use, given its context in the input sentence. It should, for example, predict the preposition "*in*" to be the most likely choice for the input sentence:

Input: *He participated* **at?** *the competition.*
Corrected: *He participated* **in** *the competition.*

If the predicted preposition differs from the original one, a *confidence* model would then need to decide whether to suggest the correction to the user. In this case, confidence in the predicted preposition "*in*" should be much higher than the original "*at*", and correction would be warranted.

The focus of this paper is on the preposition generation task. Table 1 provides some examples. In particular, we are interested in comparing the effectiveness of different kinds of features for this task.

[1] As cited in [2].

A. Gelbukh (Ed.): CICLing 2008, LNCS 4919, pp. 643–654, 2008.

Table 1. Example sentences for preposition generation. The lexical head of the PP is in *italics* and the prepositional complement is **bolded**.

Sent #	Input Text	Output
1	Pierre Vinken *joined* the board ___ a nonexecutive **director**.	as
2	The $2.5 million Byron plant was *completed* ___ **1985**.	in
3	The average maturity for funds *open* only ___ **institutions**, ...	to
4	Newsweek announced new advertising *rates* ___ **1990**.	for

Our research question is made precise and motivated in the rest of this section. Previous work is summarized (§2) and contrasted with our proposed features (§3). After a brief description of the learning algorithm (§4), experimental results are presented (§5 and §6).

1.1 Research Question

The features considered in previous research on preposition generation may be divided into three main types. Lexical features, such as word *n*-grams within a window around the preposition; the parts-of-speech (POS) tags of these words; and syntactic features, such as the word modified by the prepositional phrase (PP), or grammatical relations between pairs of words.

Unfortunately, no direct comparison has been made between these different kinds of features. Intuitively, syntactic features should be helpful in choosing the preposition. How much gain do they offer? Does their utility vary for different kinds of PP, or depend on the size of the training set? This paper seeks to fill this gap in the literature by comparing a lexical baseline feature set with a syntactic feature set that incorporates PP attachment information.

Our key finding is that PP attachment information can improve generation performance. In a memory-based learning approach, this improvement is especially notable when the training data is sparse.

1.2 Theoretical Motivations

Linguistic analyses suggest that the attachment site of the PP, as well as the argument/adjunct distinction, play significant roles in the choice of preposition. This section provides some linguistic background to motivate our research question, and also defines some terminology to be used in the rest of the paper. The material in this section is based on [5], unless otherwise stated.

Attachment. A preposition "expresses a relation between two entities, one being that represented by the prepositional complement, the other by another part of the sentence." The *prepositional complement* is, in most cases, a noun phrase[2]. That "another part of the sentence" can be a verb-, noun- or adjectival

[2] Some prepositions function as particles in phrasal verbs, e.g., "give *up*" or "give *in*". We view these particles as part of the verb and do not attempt generation.

phrase. The PP is said to be *attached* to this phrase, and the head word of this phrase is called the *lexical head* of the PP.

For example, in sentence #1 in Table 1, the preposition "*as*" expresses the relation between the prepositional complement "*director*" and its lexical head, the verb "*joined*". Knowing that the PP is attached to "*joined*", rather than to "*board*", would clearly help predict the preposition "*as*".

Argument/Adjunct Distinction. The relevance of the lexical head for the choice of preposition may depend on its relation with the prepositional complement. One aspect of this relation is the argument/adjunct distinction. In principle, "arguments depend on their lexical heads because they form an integral part of the phrase. Adjuncts do not."[6]. The preposition in an argument PP is thus more closely related to the lexical head than one in an adjunct PP. The distinction can be illustrated in two of the syntactic functions of PPs:

- **Complementation:** The preposition marks an argument of the lexical head. The prepositions "*as*" in sentence #1 in Table 1 is such an example. In this usage, the PP is said to be an *argument*.
- **Adverbial:** The PP serves as a modifier to its lexical head. The phrase "*in 1985*" in sentence #2 is one example. In this usage, the PP is an *adjunct*.

The argument/adjunct distinction has been shown to be helpful in PP attachment [6]; it may also be relevant in preposition generation.

1.3 Practical Motivations

In addition to the linguistic motivations discussed above, the use of PP attachment and the argument/adjunct distinction can also improve the user experience of a grammar checking system.

For a language learner, the system should serve not merely a practical, but also an educational, purpose. Besides having a wrong preposition detected and corrected, the user would also like to learn the reason for the correction, such as, "the verb X requires the preposition Y". Without considering PP attachment, this kind of feedback is difficult.

By making known its assumptions on the attachment site, the grammar checker also enhances its transparency. If the user spots an attachment error, for example, s/he may choose to inform the system and can then expect a better prediction of the preposition.

2 Previous Work

Previous research on preposition generation and error detection has considered lexical, part-of-speech (POS) and syntactic features.

2.1 Lexical and POS Features

A rule-based approach using lexical features is employed in [3] for Swedish prepositions. The system can identify insertion, deletion and substitution errors, but does not offer corrections.

A variety of lexical and POS features, including noun and verb phrases in the vicinity of the preposition, as well as their word lemmas and POS tags, are utilized in [2]. The evaluation data consist of newspaper text and a corpus of essays written by 11th and 12th grade students, covering 34 prepositions. A maximum entropy model achieved 69% generation accuracy. Differences in the data set genre, however, prevent a direct comparison with our results.

2.2 Syntactic Features

To our best knowledge, the only work on preposition generation that utilizes syntactic features is [4]. In addition to a variety of POS features and some WordNet categories, it also considers grammatical relations (e.g., direct or indirect object) extracted from a parser. The grammatical relation feature is identified as a strong feature. A voted perceptron algorithm, trained on five prepositions, yielded 75.6% accuracy on a subset of the British National Corpus.

3 Features

Despite the variety of features explored in previous work, no analysis on their relative effectiveness has been performed. The main goal of this paper is to make a direct comparison between lexical and syntactic features. We thus propose two feature sets, LEXICAL and ATTACH. They are restricted to the same types of features except one difference: the former contains no information on the PP attachment site; the latter does. Some examples of these features are given in Table 2.

Table 2. Two sets of features are to be contrasted. The LEXICAL feature set does not specify the PP attachment site; the ATTACH set does so via the Lexical Head feature. Features extracted from the sentences in Table 1 are shown below.

Sent #	LEXICAL			ATTACH		
	VP Head (`V`)	NP/ADJP Head (`N1`)	Complement (`N2`)	Lexical Head (`H`)	NP/ADJP Head (`N1`)	Complement (`N2`)
1	joined	board	director	joined	board	director
2	completed	*null*	1985	completed	*null*	1985
3	*null*	open	institutions	open	*null*	institutions
4	announced	rates	1990	rates	*null*	1990

3.1 Lexical Feature Set

Three words in the vicinity of the preposition are extracted[3]:

– **Verb Phrase Head** (`V`) Head of the verb phrase preceding the preposition.

[3] We follow the naming convention in the literature on PP attachment disambiguation (e.g., [7]). Our LEXICAL feature set is similar to theirs, with one crucial difference: the preposition *itself* is not included as a feature here, for obvious reasons.

- **Noun or Adjectival Phrase Head** (`N1`) Head of the noun phrase or adjectival phrase occurring between `V` and the preposition.
- **Prepositional Complement** (`N2`) Head of the noun phrase or nominal *-ing* following the preposition.

For example, for sentence #1 in Table 2, `V` is "*joined*", `N1` is "*board*", and `N2` is "*director*".

Since the PP may be attached to `V` or `N1`, its attachment site cannot be inferred from this feature set. However, either `V` or `N1` can be missing; for example, in sentence #2, `N1` is *null* because the verb "*completed*" is immediately followed by the PP "*in 1985*". In such a case, then, there is no PP attachment ambiguity.

3.2 Attachment Feature Set

In the LEXICAL feature set, the PP attachment site is left ambiguous. We hypothesize, on linguistic grounds presented in §1.2, that it can serve as an informative feature. To test this hypothesis, the ATTACH feature set re-labels the features in LEXICAL based on the PP attachment site given by the parse tree:

- **Lexical Head** (`H`) If the PP is attached to a verb phrase, the lexical head is `V`; if the PP is attached to a noun- or adjectival phrase, it is `N1`.
- **Noun or Adjectival Phrase Head** (`N1`) Similarly, this could be one of two values. If the PP is attached to a noun- or adjectival phrase, this is *null*; if it is attached to a verb phrase, this is the same as the `N1` in LEXICAL. In the latter case, the noun may still play an important role in the choice of preposition. Consider the expressions "*keep the pressure* **on** *someone*" and "*keep pace* **with** *someone*". Under the same lexical head "*keep*", the `N1` nouns "*pressure*" and "*pace*" provide strong clues about the different prepositions.
- **Prepositional Complement** (`N2`) Same as in the LEXICAL feature set.

4 Memory-Based Learning

The memory-based learning framework has been shown to perform well on a benchmark of language learning tasks [8]. In this framework, feature vectors are extracted from the training set and stored as a database of instances, called the *instance base*. For each test instance, the set of nearest neighbors is retrieved from the instance base. The majority label of this set is returned.

One strength of this approach is that irregular and low-frequency events are preserved in the instance base. This may prove advantageous for our task, as the choice of preposition can be highly context-specific and idiosyncratic.

Of critical importance is the distance metric between two instances, since it determines who the nearest neighbors are. We utilized IB1-IG [9], an algorithm that uses information gain to define this metric. The following section is a brief summary taken from [8].

4.1 IB1-IG

When there are n features, the distance Δ between two instances X and Y is:

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i)$$

where δ, the distance per feature, is defined by:

$$\delta(x_i, y_i) = \begin{cases} 0 \text{ if } x_i = y_i \\ 1 \text{ otherwise} \end{cases}$$

The weight w_i is intended to reflect the salience of the feature i. In IB1-IG, w_i is the information gain (IG) of feature i, i.e. the amount of entropy (H) reduced by the feature. In order not to favor features with more values, the "split info" ($si(f)$) is used as a normalizing factor. Formally,

$$w_i = \frac{H(C) - \sum_{v \in V_f} P(v) H(C|v)}{si(f)}$$

$$si(f) = -\sum_{v \in V_f} P(v) \log_2 P(v)$$

where C is the set of class labels (i.e., the prepositions), and V_f is the set of values for feature f.

4.2 Example

The distance metric could be understood as defining "buckets" of neighbors for each test instance. These buckets, from the nearest ones to the furthest, form the steps of the back-up sequence to be followed by the algorithm, as it searches for the set of nearest neighbors. As an illustration, we now apply the IB1-IG algorithm to the LEXICAL feature set (see §3.1).

The information gain of each feature in consideration, `V`, `N1` and `N2`, is computed on the training set. The information gain for `N1` turns out to be the greatest, followed by `N2` and then `V`. By linguistic intuition, `N1` and `N2` should be most informative for preposition generation when the lexical head is a noun. Since nouns constitute the majority among the lexical heads in our training set (see §5), it is natural that `N1` and `N2` yield the most information gain.

Table 3 shows the complete back-off sequence. Given a test instance, its closest neighbors are those training instances that match all three features (`N1+N2+V`). If such instances exist, the majority label (preposition) of these neighbors is returned. Among our test data whose lexical heads are nouns, 1111 fall into this category, and the predicted preposition is correct 78.1% of the time.

If no training instances match all three features, then the algorithm searches for training instances that match both `N1` and `N2` (`N1+N2`), since this combination yields the next largest information gain. The process continues down the back-off sequence in the left column of Table 3.

Table 3. The back-off order of the nearest-neighbor "buckets" in the LEXICAL feature set. The size of each bucket and its corresponding accuracy are listed below for two types of lexical heads: nouns, and verbs with argument PPs.

Nearest Neighbor Back-off Sequence	Lexical Head			
	Noun		Verb (argument PP)	
	Size	Accuracy	Size	Accuracy
`N1+N2+V`	1111	78.1%	395	82.3%
`N1+N2`	621	68.4%	243	24.3%
`N1+V`	471	57.5%	45	51.1%
`N2+V`	35	54.3%	14	78.6%
`N1`	14	21.4%	3	0%
`N2`	0	n/a	0	n/a
`V`	2	100%	0	n/a
Total	2254	71.8%	700	59.7%

5 Data

We restrict our attention to the ten most frequently occurring prepositions in the Penn Treebank [10]: *of, in, to, for, on, by, at, with, from*, and *as*.

Our test data consists of 3990 occurrences[4] of these ten prepositions in section 23 of the Penn Treebank. Statistics of the test data are presented in Table 4.

Table 4. Distribution of lexical heads in our test set, section 23 of the Penn Treebank

Lexical Head	Percentage
Verb (argument PP)	17.5%
Verb (adjunct PP)	22.9%
Noun	56.5%
Adjective	3.0%

Our training data is the AQUAINT Corpus of English News Text, which consists of 10 million sentences drawn from New York Times, Xinhua News Service, and the Associated Press. Parse trees for these sentences are obtained automatically from a state-of-the-art statistical parser [11]. The distributions of the prepositions are shown in Table 5.

Correctness of the PP attachment in the training data could have been ensured by using a manually parsed corpus, such as the Penn Treebank. However, the parser is reasonably accurate with PP attachments[5], and allows us to take statistical advantage of a much larger training corpus such as AQUAINT. This advantage is especially significant for the memory-based learning framework. Our

[4] Some prepositions occur in constructions such as "**as ... as**", "*because* **of**" and "*such* **as**", where their usage is quite predictable. To avoid artificially boosting the generation accuracy, we exclude such cases from our experiments.

[5] The parser achieves 82.29% recall and 81.51% precision [11] for PP modifications.

Table 5. The five most frequently occurring prepositions in the training set, tabulated according to their lexical heads

Verbs		Nouns		Adjectives	
Prep.	Frequency	Prep.	Frequency	Prep.	Frequency
in	25%	*of*	55%	*to*	27%
to	16%	*in*	15%	*of*	14%
for	11%	*for*	8%	*for*	14%
on	10%	*on*	5%	*as*	13%
with	10%	*to*	4%	*with*	11%

results may also be more realistic, since treebanks may not be available in other domains.

6 Evaluation

We conducted experiments to compare the two feature sets described in §3: Lexical and Attach. Results are summarized in Table 6.

Table 6. Preposition generation accuracy on the Lexical and Attach feature sets. The majority baseline is 28.5% (always choosing "*of*"). Results from §6.2 are upper bound estimations; results from §6.4 is our best without assuming correct attachment information in the test input. For detailed comments, see the individual sections listed in the left column.

Section	Train Set	Test Set	Verbs (argument PP)	Verbs (adjunct PP)	Nouns	Adjectives	Overall
§6.1	Lexical	Lexical	59.7%	58.6%	71.8%	75.8%	66.8%
§6.4	Attach	Lexical	**72.3%**	**60.2%**	**71.7%**	**77.5%**	**69.3%**
§6.2	Attach	Attach	75.3%	62.8%	72.5%	81.7%	71.1%
§6.2	Attach	Arg	75.3%	65.9%	n/a	n/a	n/a

6.1 Lexical Feature Set

As discussed in §4.2, information gain is greatest for the NP/ADJP Head feature (`N1`) in the Lexical feature set, followed by Prepositional Complement (`N2`), and lastly Verb Phrase Head (`V`). This sequence produces the back-off steps of nearest neighbors shown in Table 3. Please refer to this table for the rest of this section.

Nouns and Adjectives. When very similar training instances (`N1+N2+V`) are available, generation accuracy reaches a relatively high 78.1%. Performance gradually degrades as the nearest neighbors become less similar. The overall accuracy is 71.8% for nouns. The same general trend is observed for adjectives.

Verbs. Our discussion on verbs will focus on those with argument PPs. Generation accuracy is relatively high (82.3%) when similar neighbors (`N1+N2+V`) are available. However, at the next back-off level, `N1+N2`, the accuracy sharply decreases to 24.3%. This drags the overall accuracy down to 59.7%.

The poor performance when backing off to `N1+N2` is not accidental. The VP Head (`V`) feature is most relevant when an argument PP is attached to a verb. Consider the sentence "*They've never shown any inclination to spend money* **on** *production*". Among the `N1+N2` neighbors, the preposition "*for*" is the most common, due to expressions such as "*money* **for** *production*". However, the verb "*spend*", coupled with a direct object "*money*", should have signaled a strong preference for the preposition "*on*".

In other words, backing off to `V+N2` would have been more appropriate, since the word "*production*" is related more to the verb than to the `N1` noun. An obvious remedy is to use a different back-off sequence when the lexical head is a verb. However, there is no way of making this decision, precisely because the PP attachment site is not known. The Attach feature set is designed to address this shortcoming.

6.2 Attachment Feature Set: With Treebank

Without the benefit of attachment information, the Lexical feature set is limited to one back-off sequence, ignoring the underlying differences between PPs with verb and noun lexical heads. In contrast, the Attach feature set creates an instance base for each kind of lexical head. Each instance base can then optimize its own back-off sequence.

Performance of the Attach feature set depends critically on the quality of the PP attachment information. We therefore performed evaluation on the test set under three conditions. In this section, the features were extracted from the manually parsed Penn Treebank; in §6.3, they were extracted from automatically produced parse trees; in §6.4, no parse tree was assumed to be available.

Nouns and Adjectives. Information gain is greatest for Lexical Head (`H`), then Prepositional Complement (`N2`). Accuracies for both nouns and adjectives (third row in Table 6) compare favorably with the Lexical set, likely due to the fact that `N2` counts are no longer skewed by verb-specific usage.

Verbs. Information gain is highest for `H`, followed by `N2` and `N1`, yielding the back-off order shown in Table 7. Generation accuracy is 75.3% for verbs with argument PPs, substantially higher than the Lexical feature set, at 59.7%.

For those test instances with very similar training counterparts (`H+N1+N2`), the accuracy is 82.3%. This performance is comparable to the analogous category (`N1+N2+V`) in the Lexical feature set. The gain over the Lexical feature set is mainly due to the appropriate back-off to `H+N2`, which yields 66.4% accuracy. This back-off decision, in contrast to the one with the Lexical set, recognizes the importance of the identity of the verb.

Overall, when assuming perfect attachment information, the generation accuracy for the Attach feature set is 71.1% (third row in Table 6).

Table 7. Back-off order of the nearest-neighbor "buckets" for verb lexical heads in the Attach feature set. Performance of verbs with argument PPs are listed.

Nearest Neighbor	Verb (argument PP)	
Back-off Sequence	Size	Accuracy
`H+N1+N2`	389	82.3%
`H+N2`	143	66.4%
`H`	167	67.1%
`N2`	1	0%
Total	700	75.3%

Argument/Adjunct Distinction. For verbs[6], further gain in accuracy is still possible if the argument/adjunct distinction is known. Preposition generation tends to be more difficult for verbs with adjunct PPs than those with argument PPs. Since adjuncts depend less on the verb than arguments, their performance naturally suffers at the back-off to `H`. At this back-off level, arguments achieve 67.1% accuracy (see Table 7). The analogous figure for adjuncts is only 31.8%.

One case in point is the sentence "*... other snags that infuriated some fund investors* **in** *October 1987*". As an adjunct, the preposition "*in*" should be highly likely in front of the word "*October*". The back-off to `H`, however, wrongly predicts "*by*" based on statistics associated with the verb "*infuriated*".

Suppose the argument/adjunct distinction is known in the test data, and that the back-off from `H+N2` is changed from `H` to `N2` when the PP is an adjunct. The performance for adjuncts would then rise to 65.9% (last row in Table 6), an absolute improvement of 3%.

6.3 Attachment Feature Set: With Automatically Derived Parse Trees

In the previous section, where perfect attachment information is available, the overall generation accuracy reaches 71.1%. This section considers the use of automatically parsed sentences [11] rather than the Penn Treebank. The result should still be interpreted as an upper bound, since the parsing was performed on sentences with the correct prepositions in place.

When the Attach features are extracted from these parse trees, the overall generation accuracy decreases to 70.5%. It would be interesting to observe how much further the accuracy would degrade if sentences with preposition errors are fed to the parser. Making a meaningful comparison might prove difficult, however, since one needs to simulate how the test sentences would have been written by non-native speakers of English.

Instead, we now discuss some techniques which, without relying on attachment annotation in input sentences, could still help improve the accuracy.

[6] We consider verbs only, since "it is difficult to consistently annotate an argument/adjunct distinction" [12] for nouns in the Penn Treebank.

6.4 Attachment Feature Set: Without Parse Trees

For texts with lots of grammatical errors, parsing could be challenging, making it difficult to obtain attachment information. Lexical features, however, can be extracted more robustly. Could test instances with only LEXICAL features still leverage an instance base with ATTACH features?

A significant portion of prepositional phrases, in fact, have no ambiguity in their attachment site; for example, when a verb is immediately followed by a preposition, or when an `N1` noun occurs at the beginning of the sentence. The unambiguous test instances, then, can take advantage of the ATTACH instance base, while the rest are processed as usual with the LEXICAL instance base. This simple mechanism improves the overall accuracy from 66.8% to 68.7%.

For the ambiguous instances[7], their performance on the LEXICAL instance base still has room for improvement. As we have seen in §6.1, the back-off decision is crucial when fully matched instances (`N1+N2+V`) are not available. Instead of always backing off to `N1+N2`, entropy statistics can help make more informed choices.

Three sets of nearest neighbors — `N1+N2`, `N1+V` and `N2+V` — are the back-off options. If the lexical head is a verb, for example, one may expect the back-off sets involving `V` to have relatively low entropy, since the distribution of their prepositions should be more constrained. One reasonable approach is to back-off to the set with the lowest entropy. This procedure raises the overall accuracy to 69.3% (second row in Table 6), which is within 2% of the upper bound.

7 Conclusions

We have shown that knowledge of the PP attachment site can improve accuracy in preposition generation. In a memory-based learning framework, the improvement is especially substantial when similar training instances are not available and a back-off decision must be made.

For noisy texts, such as input to a grammar checker, PP attachment sites may not be readily available. In these cases, attachment information in training data can still boost generation accuracy to within 2% of the upper bound.

Acknowledgments

We thank Viggo Kann, Stephanie Seneff, Martin Volk, as well as the two anonymous reviewers, for their helpful discussions.

[7] Another potential approach is to first disambiguate the PP attachment site, i.e., determine whether `V` or `N1` is to be assigned as the lexical head `H`. The instance base with ATTACH features can then be used as before. We have not explored this approach, since literature on PP attachment disambiguation suggests that the preposition identity is one of the most important features [13].

References

1. Bitchener, J., Young, S., Cameron, D.: The effect of different types of corrective feedback on esl student writing. Journal of Second Language Writing 14, 191–205 (2005)
2. Chodorow, M., Tetreault, J., Han, N.R.: Detection of grammatical errors involving prepositions. In: Proc. ACL-SIGSEM Workshop on Prepositions, Prague, Czech Republic (2007)
3. Eeg-Olofsson, J., Knutsson, O.: Automatic grammar checking for second language learners — the use of prepositions. In: Proc. NoDaLiDa, Reykjavík, Iceland (2003)
4. De Felice, R., Pulman, S.G.: Automatically acquiring models of preposition use. In: Proc. ACL-SIGSEM Workshop on Prepositions, Prague, Czech Republic (2007)
5. Quirk, R., et al.: A comprehensive grammar of the english language. Longman (1985)
6. Merlo, P., Esteve Ferrer, E.: The notion of argument in prepositional phrase attachment. Computational Linguistics 32, 341–378 (2006)
7. Ratnaparkhi, A., Reynar, J., Roukos, S.: A maximum entropy model for prepositional phrase attachment. In: Proc. ARPA Workshop on Human Language Technology, Plainsboro, NJ (1994)
8. Daelemans, W., van den Bosch, A., Zavrel, J.: Forgetting exceptions is harmful in language learning. Machine Learning 34, 11–41 (1999)
9. Daelemans, W., van den Bosch, A., Weijters, A.: Igtree: Using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review 11, 407–423 (1997)
10. Marcus, M., et al.: The penn treebank: Annotating predicate argument structure. In: Proc. ARPA Workshop on Human Language Technology, Plainsboro, NJ (1994)
11. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics 29, 589–637 (2003)
12. Bies, A., et al.: Bracketing guidelines for treebank ii style. Technical Report, University of Pennsylvania, Philadelphia, PA (1995)
13. Collins, M., Brooks, J.: Prepositional phrase attachment through a backed-off model. In: Proc. 3rd Workshop on Very Large Corpora, Cambridge, MA (1995)