Word Clustering for Persian Statistical Parsing

Masood Ghayoomi

German Grammar Group, Freie Universität Berlin, Germany
masood.ghayoomi@fu-berlin.de

Abstract. Syntactically annotated data like a treebank are used for training the statistical parsers. One of the main aspects in developing statistical parsers is their sensitivity to the training data. Since data sparsity is the biggest challenge in data oriented analyses, parsers have a malperformance if they are trained with a small set of data, or when the genre of the training and the test data are not equal. In this paper, we propose a word-clustering approach using the Brown algorithm to overcome these problems. Using the proposed class-based model, a more coarser level of the lexicon is created compared to the words. In addition, we propose an extension to the clustering approach in which the POS tags of the words are also taken into the consideration while clustering the words. We prove that adding this information improves the performance of clustering specially for homographs. In usual word clusterings, homographs are treated equally; while the proposed extended model considers the homographs distinct and causes them to be assigned to different clusters. The experimental results show that the class-based approach outperforms the word-based parsing in general. Moreover, we show the superiority of the proposed extension of the class-based parsing to the model which only uses words for clustering.

Keywords: Statistical Parsing, Word Clustering, the Persian Language.

1 Introduction

Parsing a natural language aims to provide a syntactic analysis of a sentence. To achieve this goal automatically, a parser, either rule-based or statistical, should be used. Data oriented parsers are trained with annotated data, like a treebank. Contrary to rule-based parsers, the statistical parsers are very sensitive to the data they are trained with, and one big problem of the training data is that it is always sparse. As a result, it is very difficult to build an accurate model from sparse data. Additionally, it is very likely to face unknown words while parsing in real applications.

Word clustering has caught attention in natural language processing to represent a coarser level of the lexical information rather than the words themselves. In this approach, words are clustered in an off-line process based on their occurrence in an unannotated corpus through an unsupervised method. In our study, we aim to use a word clustering approach for parsing to improve the performance of our statistical parser for Persian trained with a very small amount of

H. Isahara and K. Kanzaki (Eds.): JapTAL 2012, LNAI 7614, pp. 126–137, 2012.

data. One important problem of word clustering is that homographs are treated equally which leads them to be clustered inaccurately. In addition to the class-based parsing, we propose a model which uses the part-of-speech (POS) tags of the words as an important additional lexical information in clustering to distinct the homographs and to cluster them into different classes consequently.

The structure of this paper is as follows: in Section 2 we briefly describe some basic properties of Persian. In Section 3, the tool used for parsing Persian is explained. Section 4 devotes to the treebank used for training the parser. Section 5 describes class-based parsing and the Brown algorithm used for this aim. Section 6 explains the setup of the experiments for the proposed parsing models and the obtained results; and finally, the paper is summarized in Section 7.

2 The Persian Language

Persian is a member of the Indo-European language family and it has many features in common with the other languages of this family in terms of phonology, morphology, syntax, and lexicon. Persian uses a modied version of the Arabic script and it is written right-to-left. However, the two languages differ from one another in many respects. Persian belongs to the subject-drop languages with an SOV constituent order in unmarked constructions. The constituent order is relatively free. Verbs are inflected for tense and aspect, and they agree with the subject in person and number. The language does not make use of gender [19]. There exists a so called ‘pseudo-space’ in the internal structure of the Persian lexical items. Using a white space rather than ‘pseudo-space’ will intensify the multi-word token problem. Moreover, contrary to long vowels, short vowels usually are not written but they are pronounced. This property leads to have more homographs in written texts.

3 Stanford Parser for Persian

The Stanford parser is a Java implementation of a lexicalized, probabilistic natural language parser [14]. The parser is based on an optimized Probabilistic Context Free Grammar (PCFG) and lexicalized dependency parsers, and a lexicalized PCFG parser. The output of the parser provides the phrase structure tree of a sentence along with the dependencies of the words in the sentence.

Three basic modules, namely *FactoredLexicon*, *ChineseLexicon*, and *BaseLexicon* modules, are defined in the parser for learning the lexicon. The most important task of these modules is to calculate the probability of a word given its tag, $P(word|tag)$, to let the parser choose the best tag of the word in the local context, and to utilize this probability to find the best tree structure for a sentence. The *BaseLexicon* module learns the lexicon from the training data, say the treebank. This module has worked quite appropriately for Penn English Treebank. In the adaptation of the Stanford parser for Persian, we have used the *BaseLexicon* module as well.

It should be added that a morphological tokenizer and lemmatizer are required within the Stanford CoreNLP package. Since this package currently lacks these tools for Persian, the parser assumes that tokenization and lemmatization have already been done both on the training data and test data. However, these shortcomings may affect the performance of the parser in real applications.

Following the study of Collins [9], to make the parser able to work with a data from a treebank, it is required to provide the list of heads in the phrase structure trees. To define the heads semi-automatically for Persian, we extracted all the grammar rules from the Persian treebank (PerTreeBank) and based on the labels of the mother nodes, we determined the heads of the constituents for the parser.

4 The Persian Treebank

PerTreeBank [1] is the first treebank for Persian which is developed in the framework of the HPSG [23] formalism and it is freely available on-line. No feature structures are used in the development of this treebank, but basic properties of HPSG are simulated. This treebank contains 1012 trees from the Bijankhan Corpus[2] and it is developed semi-autmatically via a bootstrapping approach [11,12]. This treebank which has the XML data structure provides the phrase structure trees of the sentences in the Chomskyan grammar such that the type of the dependencies in the nodes' relations of the mother nodes are defined explicitly according to the basic schemas in HPSG, namely head-subject, head-complement, head-adjunct, and head-filler to bind off the extraposed constituents. It needs to be added that the canonical positions of the scrambled or extraposed elements are explicitly determined with the *nid* (not immediate dominance) node; therefore trace-based analyses of sentences are provided in PerTreeBank. Moreover, elliptical elements are also determined explicitly with a node which defines the type of ellipsis. The available morpho-syntactic and semantic information of the words in the Bijankhan Corpus is also used for the words of the treebank; as a result, the treebank is rich both in terms of the available information of the POS tags and the tree analyses of the sentences. Figure 1 displays the tree representation of sample (1):

(1) born be donbāle in ast ke čizi rā besāzad
Born to follow.EZ this is that something.RES DOM SUBJ.create.3SG
ke qablan vojud nadāšteast.
that before existence NEG.had.CL.3sg
'Born is after this [namely] to create something that did not exist before.'

To use the treebank for our experiments, we need to normalize the trees and convert the treebank from the XML format into a plain text Penn Treebank style. To this end, several conversion is done on the treebank. As said, Persian is

[1] http://hpsg.fu-berlin.de/~ghayoomi/PTB.html
[2] http://ece.ut.ac.ir/dbrg/bijankhan/

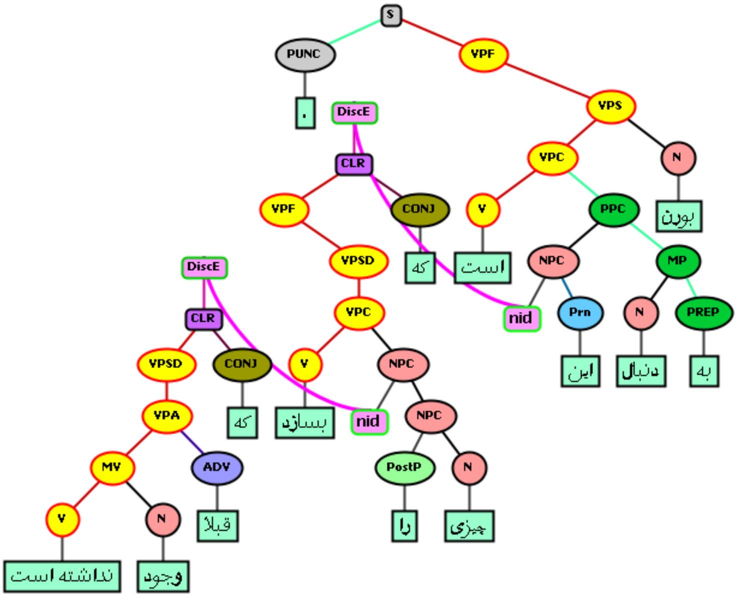


Fig. 1. Right-to-left tree representation of example (1)

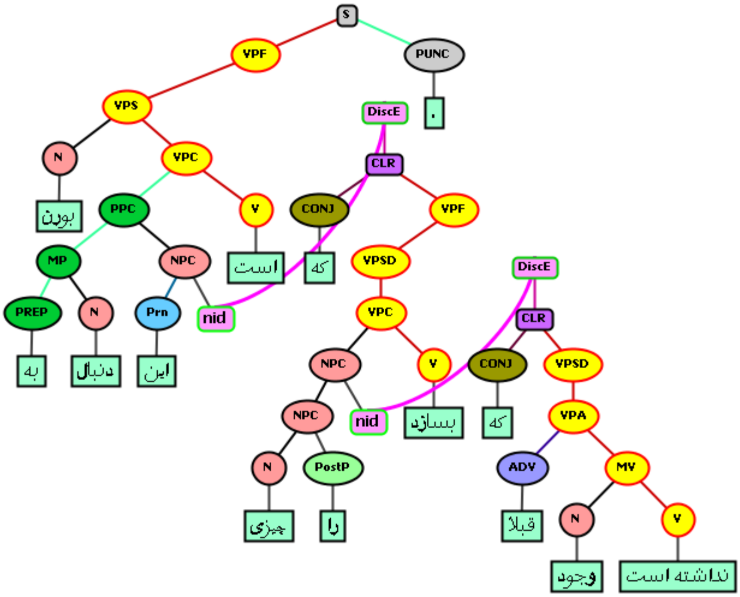


Fig. 2. Left-to-right tree representation of example (1)

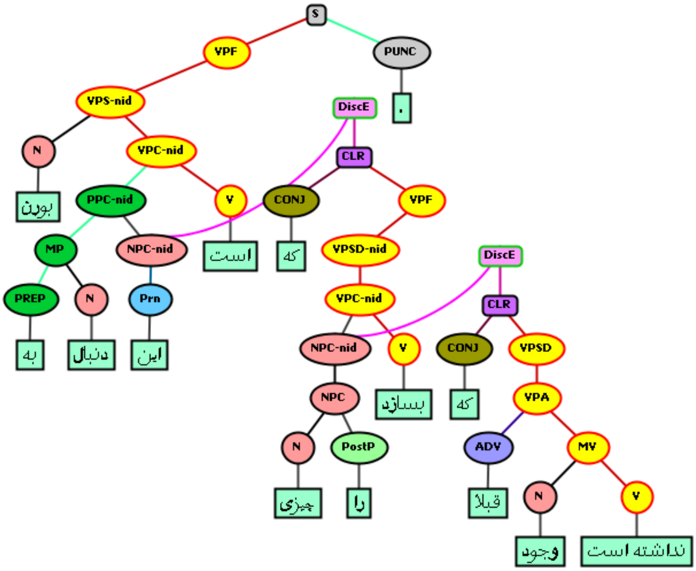
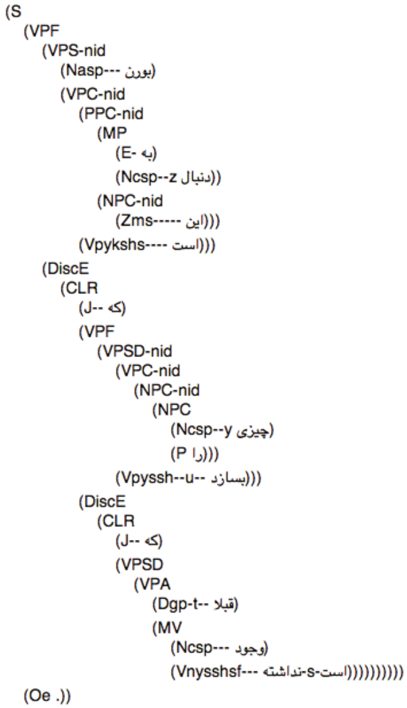


Fig. 3. Traceless left-to-right tree representation of example (1)



```
(S
  (VPF
    (VPS-nid
      (Nasp--- بورن)
      (VPC-nid
        (PPC-nid
          (MP
            (E- به)
            (Ncsp--z دنبال))
          (NPC-nid
            (Zms----- این)))
        (Vpykshs---- است)))
    (DiscE
      (CLR
        (J-- که)
        (VPF
          (VPSD-nid
            (VPC-nid
              (NPC-nid
                (NPC
                  (Ncsp--y چیزی)
                  (P را)))
              (Vpyssh--u-- بسازد)))
        (DiscE
          (CLR
            (J-- که)
            (VPSD
              (VPA
                (Dgp-t-- قبلا)
                (MV
                  (Ncsp--- وجود)
                  (Vnysshsf--- نداشته-s-است))))))))))
  (Oe .))
```

Fig. 4. Penn style tree representation of example (1)

a right-to-left language. Since the Stanford parser does not support bidirectional parsing, we have to convert the treebank into left-to-right direction, similar to Penn Arabic Treebank[3], without loosing any information as it is displayed in Figures 2. Additionally, since we want to train the parser with trace-less trees, the *nid* nodes should be removed. Before then, the mother nodes of the *nid* nodes are renamed as X−nid, and '−nid' which functions as a slashed element in HPSG is propagated to the node where it is bound off by head-filler schema as represented in Figure 3. After converting the XML format of the trees into a plain text Penn Treebank format, which is demonstrated in Figure 4, the trace-less, Penn-Treebank-style data is used to train the Stanford parser for Persian. Moreover, in the normalization process, the following information is also lost from the original treebank: structure sharing, the links of the extraposed or scrambled elements to their corresponding canonical positions, the *Pragmatic* node, the named entities tags, the lemmas, and the types of /ke/, /ye/, and clitics.

After converting the original data into the Penn Treebank format, we use the *PennTreebankTokenizer* module in the Stanford parser to tokenize the input data. It is presumed that the input data is properly tokenized with a white-space. Sine there is a possibility to use a white-space or pseudo-space between the elements of a word, it is replaced by '−s−' in the internal structures of the lexical items to recognize multi-tokens as one unit and to solve the problem of tokenization.

We need to point out that even though the Stanford parser is the implementation of PCFG and the Persian data set used for training the parser is HPSG-based, there is no conflict between them, since the trees look like the phrase structure trees. Moreover, training a PCFG parser with an HPSG treebank is also experimented in other studies [27].

5 Class-Based Parsing

Brown [4] was the pioneer to use word clustering for language modeling methods. Later on, word clustering has been widely used in various natural language processing applications including parsing [5,6,7,16], word sense disambiguation [18], automatic thesaurus generation [13], machine translation [26], sentence retrieval [21], named entity tagging [20], language model adaptation [15], speech recognition [24], query expansion [1], and text categorization [8].

Using word clustering has advantages and disadvantages. One of the advantages of word clustering is reducing the data sparsity problem. Hence, if the word is not seen but its class, then the performance of the system will not be reduced due to the out of vocabulary problem. This approach is very effective, specially when the genre of the data changes. Another advantage of word clustering is its flexibility to capture different features. For example, semantic or syntactic properties of words can be captured using different word clustering algorithms. Since our aim for statistical parsing is to group the words with similar

[3] http://www.ircs.upenn.edu/arabic/

syntactic behavior, this flexibility gives us the opportunity to choose a sophisticated algorithm which captures the syntactic similarities of words to be used for parsing. The disadvantage of clustering is that different syntactic behaviors of homographs are not distinguished, since they are grouped in one cluster. This problem might have a counter effect for applications like parsing. Although a soft clustering approach sounds a good solution to overcome this problem, it has been shown that the overall performance of hard clustering is still better than soft clustering [10]. To resolve the problem of mis-clustering of homographs by using a hard clustering approach, we extend the word clustering algorithm by adding the POS tags of the words as an additional lexical information to the lexical items in order to recognize homographs distinctly.

Assuming that the word clustering algorithm has clustered the words of a text accurately, it is obvious that there is a clear relationship between the words belonging to the same cluster. The followings show some examples of word clusters created by the Brown algorithm [4] for Persian:

- CLUSTER1: *porxatartarin* [the most dangrous], *šomālitarin* [the most Northern], *zayiftarin* [the weakest], ...
- CLUSTER2: *pākizegi* [cleanness], *bastani* [ice-cream], *zibāyi* [beauty], ...
- CLUSTER3: *farmude?id* [have prescribed], *kardeast* [has done], *kardeand* [have done], ...

Such word clusters help us to find the set of terms syntactically related to each other. So that, if only one of the words of a cluster appears in the training data, the statistical parser can parse the input sentences which contain other words of the same cluster, even though these words do not exist in the training data. For example, if the word *'porxatartarin'* has been seen in the training data and it creates a noun phrase with the term *'masir'* [path], the class-based model is able to parse sentences that contain the word *'šomālitarin'* which is unseen in the test data but belongs to the same cluster as *'porxatartarin'*, and it can be combined with the term *'masir'* to create a constituent.

In the word-based scenario, the parser will be trained with the treebank containing the words with their corresponding POS tags, and the syntactic annotations. In the class-based approach, the words should be clustered into a set of predefined number of clusters. Having a mapping between the words and their corresponding clusters, the parser is trained with word clusters in the treebank instead of the words themselves.

5.1 The Brown Clustering Algorithm

Brown clustering [4] is a hierarchical bottom-up algorithm which uses Average Mutual Information (AMI) between the adjacent clusters to merge cluster pairs. Using AMI, the algorithm considers the context information to find the similar words and put them in the same cluster. To this aim, a set of word bigrams $f(w, w')$ from an input corpus is required, where $f(w, w')$ is the number of times the word w' is seen in the context w. Both w and w' are assumed to come from a common vocabulary. Using this algorithm for clustering words, different words seen in the same contexts will be merged, because appearing in the same context shows that these words can be replaced by each other and they are assigned to the same cluster as a result [22].

One of the advantages of the Brown algorithm is using mutual information as a similarity measure. Since word bigram statistics are useful for syntax similarity, this model can be used for clustering in parsing. The mutual information of the two adjacent clusters $(C_w, C_{w'})$ is calculated as follows:

$$MI(C_w, C_{w'}) = \log \frac{P(C_w, C_{w'})}{P(C_w) * P(C_{w'})}$$

If w' follows w less often than we expect on the basis of their independent frequencies, then the mutual information is negative. If w' follows w more often than we expect, then the mutual information is positive [4]. Algorithm 1 shows the Brown word clustering algorithm in more detail.

Algorithm 1. The Brown Word Clustering Algorithm

```
Initial Mapping: Put a single word in each cluster
Compute the initial AMI of the collection
repeat
    Merge the pair of clusters which has the minimum decrement in AMI
    Compute the AMI of the new collection
until reach the predefined number of clusters
repeat
    Move each word to the cluster that offer the highest AMI
until no change is observed in AMI
```

As shown in the algorithm, clusters are initialized with a single term in each cluster. Then, in each iteration, the best cluster pair, which offers a minimum decrement in AMI, is combined together. The process continues for $V - K$ iterations, where V is the number of terms and K is the predefined number of clusters. In the final step after the iterative process, all words are temporary moved from one cluster to the other cluster one by one, and AMI is recalculated. If this reassignment increases AMI, then the word will be moved to a cluster which offers the highest AMI. The algorithm is stopped when no additional increment in AMI is observed [4].

5.2 Word Representation for Clustering

As described in the previous section, the Brown algorithm originally used the word bigrams from a raw corpus for clustering (thereafter we call it Model A). The output of the clustering is hard; i.e. each lexical item is assigned to only one cluster. The advantage of this clustering is reducing the data sparsity which has a positive impact on statistical parsing. However, the main shortcoming of hard clustering is restricting each lexical item to one class which is not ideal for homographs. This problem is more pronounced for Persian text processing since short vowels are not written. Bijankhan et al. [3] have defined syntactic patterns to distinguish Persian homographs; therefore, we used the POS tags of the words to disambiguated a large portion of homographs for clustering (thereafter we call it Model B). As an example, the string 'š.v.m' could be either pronounced /šum/ as an adjective which means 'evil' or /šavam/ as a verb which means 'become.1SG'. Using the normal word clustering, these two words are treated equally, and they are assigned to only one cluster. While in the extended version, the main POS tag of the word is used as an additional lexical information for clustering; as a result, the homographs which have different POS tags are assigned to different clusters, in case they have different POS tags. To prepare the input corpus for the extended model as the input data to the Brown algorithm, the POS tag of the word is joined to the word with a hyphen, like: 'šum-ADJ', and 'šavam-V'.

6 Evaluation

6.1 Setup the Experiments

Clustering Tool. Before evaluating the class-based model, we had to cluster lexical items by the Brown word clustering algorithm described in Section 5. To this aim, we used the SRILM toolkit [25] as it contains the implementation of the Brown algorithm.

Clustering Data Set. To set up the experiments of our proposed models for parsing, we used the Bijankhan Corpus for both models of clustering. The Bijankhan Corpus is a sub-corpus of Peykare, a big balanced corpus for Persian [2,3]. The Bijankhan Corpus contains more than 2.5 million word tokens, and it is POS tagged manually with a rich set of 586 tags containing morpho-syntactic and semantic information. Following the EAGLES guidelines [17], there is a hierarchy on the assigned tags such that the first tag expresses the main syntactic category of the word followed by a set of morpho-syntactic and semantic features. The main POS tag of the word in the Bijankhan Corpus which is a set of 14 labels is used for distinguishing homographs.

6.2 Results

To evaluate the performance of the Stanford parser for Persian based on our models, the parser is trained with PerTreeBank represented by either words or clusters. For class-based models (Models A and B), the treebank is converted in such a way that the words of the treebank are mapped to the clusters in Model A, and again the words of the treebank are mapped to the clusters with respect to their POS tags in Model B. Since no gold standard data is available for Persian, we used a 10-fold cross-validation to evaluate our models and study the impacts of our models on the parser's performance. As a result, 10% of the data was recognized as the test data and the rest as the training data.

Table 1. The performance of the Stanford parser for clustering parsing (Model B)

Number of Clusters	**F_1-Score**
100	55.80
500	55.59
700	**59.32**
1000	55.81

Table 2. The performance of the Stanford parser for different models of parsing

Model	**Precision**	**Recall**	**F_1-Score**
Word	50.16	49.96	50.05
Class (Model A)	58.52	58.48	58.50
Class (Model B)	59.31	59.32	**59.32**

For all the experiments, a vocabulary of 90,901 terms are used for Model A, and 98,659 terms for Model B. As the two vocabulary sizes show, around 7,758 more terms are added to the vocabulary for Model B which obviously indicates that our proposed extended model has made homographs to be distinct.

Since the Brown algorithm requires a pre-defined number of clusters, we performed our experiments on 100, 500, 700, and 1000 clusters of the vocabulary terms. Table 1 presents the performance of the class-based parsing using the Model B with different numbers of clusters. As we can see in this table, the performance of the parsing is not very sensitive to the number of clusters which shows that it is not required to fine tune the number of clusters, and we can achieve a reasonable performance by different number of clusters. Nonetheless, according to the experimental results, the best performance is achieved by clustering all vocabulary terms into 700 clusters. As a result, this number of clusters is fixed for the rest of our experiments.

Table 2 compares the results of the class-based parsing (Model A and Model B) with word-based parsing. As shown in the table, the class-based models outperform the base-line word-based model. The difference between the performance of these two models are statistically significant according to the 2-tailed t-test ($p < 0.01$). This result indicates that even though the class-based approach generalizes the word representation, it has a positive impact on the performance of statistical parsing by reducing the data sparsity and solving the out of vocabulary problem. Moreover, the proposed extension of clustering (Model B) outperforms Model A which shows that adding POS information can improve the class-based parsing result by assigning homographs to different clusters. Based on the results, resolving the problem of clustering the homographs does have a positive impact in parsing such that the achieved improvement by Model B is statistically significant ($p < 0.01$) according to the 2-tailed t-test. According to the results summarized in Table 2, we can see the same behavior on the precision and recall of the models; i.e., precision and recall of the class-based models are higher than the word-based model and Model B performs the best.

7 Summary

Statistical parsers are trained with syntactically annotated data like a treebank. Not all languages have such a rich language resource for parsing; or if one exists, it suffers from the data sparsity problem. Word clustering is a recognized method for reducing the data sparsity problem and making it genre independent, since a more coarser level of the lexicon rather than the words are created. The Brown algorithm measures the syntactic similarities in a raw text to cluster words into a pre-defined number of clusters. The result of this algorithm is a hard clustering; therefore, each word is assigned into one cluster. The problem of this clustering method is that homographs are treated equally, and they are assigned into one cluster. This problem is more pronounced in Persian in which short vowels are not written. To resolve the problem relatively, we used the POS tags of the words as an additional lexical information to differentiate the homographs. We found that the class-based parsing, in general, outperforms word-based parsing significantly. Additionally, the extended model of word clustering which uses the POS tags as an additional lexical information significantly outperforms the word clustering model which uses words only.

Acknowledgement. Masood Ghayoomi is funded by the German research council DFG under the contract number MU 2822/3-1.

References

1. Aono, M., Doi, H.: A Method for Query Expansion Using a Hierarchy of Clusters. In: Lee, G.G., Yamada, A., Meng, H., Myaeng, S.-H. (eds.) AIRS 2005. LNCS, vol. 3689, pp. 479–484. Springer, Heidelberg (2005)
2. Bijankhan, M.: The role of corpora in writing grammar. Journal of Linguistics 19(2), 48–67 (2004)
3. Bijankhan, M., Sheykhzadegan, J., Bahrani, M., Ghayoomi, M.: Lessons from building a Persian written corpus: Peykare. Language Resources and Evaluation 45(2), 143–164 (2011)
4. Brown, P.F., de Souza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational Linguistics 18, 467–479 (1992)
5. Candito, M., Anguiano, E.H., Seddah, D.: A word clustering approach to domain adaptation: Effective parsing of biomedical texts. In: Proceedings of the 12th International Conference on Parsing Technology, pp. 37–42 (2011)
6. Candito, M., Crabbe, B.: Improving generative statistical parsing with semi-supervised word clustering. In: Proceedings of the 11th International Conference on Parsing Technologies, Parise, France, pp. 138–141 (2009)
7. Candito, M., Seddah, D.: Parsing word clusters. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, Los Angeles, California, pp. 76–84 (2010)
8. Chen, W., Chang, X., Wang, H., Zhu, J., Yao, T.: Automatic Word Clustering for Text Categorization Using Global Information. In: Myaeng, S.-H., Zhou, M., Wong, K.-F., Zhang, H.-J. (eds.) AIRS 2004. LNCS, vol. 3411, pp. 1–11. Springer, Heidelberg (2005)

9. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania (1999)
10. Dhillon, I.S., Mallela, S., Kumar, R.: Enhanced word clustering for hierarchical text classification. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 191–200 (2002)
11. Ghayoomi, M.: Bootstrapping the development of an HPSG-based treebank for Persian. Linguistic Issues in Language Technology 7(1) (2012)
12. Ghayoomi, M.: From grammar rule extraction to treebanking: A bootstrapping approach. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, pp. 1912–1919 (2012)
13. Hodge, V., Austin, J.: Hierarchical word clustering - automatic thesaurus generation. Neurocomputing 48, 819–846 (2002)
14. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)
15. Kneser, R., Peters, J.: Semantic clustering for adaptive language modeling. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE Computer Society (1997)
16. Koo, T., Carreras, X., Collins, M.: Simple seim-supervised dependency parsing. In: Proceedings of the ACL 2008, Colimbus, USA, pp. 595–603 (2008)
17. Leech, G., Wilson, A.: Standards for Tagsets. In: Text, Speech, and Language Technology, 9th edn., pp. 55–80. Kluwer Academic Publishers, Dordrecht (1999)
18. Li, H.: Word clustering and disambiguation based on co-occurrence data. Natural Language Engineering 8(1), 25–42 (2002)
19. Mahootiyan, S.: Persian. Routledge (1997)
20. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: Proceedings of NAACL-HLT, pp. 337–342. Association for Computational Linguistics (2004)
21. Momtazi, S., Klakow, D.: A word clustering approach for language model-based sentence retrieval in question answering systems. In: Proceedings of the Annual International ACM Conference on Information and Knowledge Management (CIKM), pp. 1911–1914. ACM (2009)
22. Morita, K., Atlam, E.S., Fuketra, M., Tsuda, K., Oono, M., Aoe, J.: Word classification and hierarchy using co-occurrence word information. Information Processing and Management 40(6), 957–972 (2004)
23. Pollard, C.J., Sag, I.A.: Head-Driven Phrase Structure Grammar. University of Chicago Press (1994)
24. Samuelsson, C., Reichl, W.: A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE Computer Society (1999)
25. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing (ICSLP) (2002)
26. Uszkoreit, J., Brants, T.: Distributed word clustering for large scale class-based language modeling in machine translation. In: Proceedings of the International Conference of the Association for Computational Linguistics (ACL). Association for Computational Linguistics (2008)
27. Zhang, Y., Krieger, H.U.: Large-scale corpus-driven PCFG approximation of an HPSG. In: Proceedings of the 12th International Conference on Parsing Technologies, pp. 198–208 (2011)