EXPLORING CONTEXT CLUSTERING FOR TERM TRANSLATION

Alisa Zhila (alisa_zh@mail.ru)
Center for Computing Research, Instituto Politécnico Nacional, Mexico

Alexander Gelbukh (gelbukh@gelbukh.com)
Center for Computing Research, Instituto Politécnico Nacional, Mexico

Abstract Many tasks in natural language processing, such as machine translation, word sense disambiguation, word translation disambiguation, require analysis of contextual information. In case of supervised approaches this analysis is performed by human experts, which is very costly. Unsupervised approaches offer fully automatic methods to fulfill these tasks. Yet these methods are not robust, their results are very parameter-dependent and difficult to interpret. Context clustering is an unsupervised technique for analysis of context similarities. In this work we explore dependencies of context clustering results from various clustering parameters. We also explore suitability of the context clustering for word translation disambiguation by evaluating the clustering results against known classes that are classes of translation candidates.

Keywords translation, translation candidates, clustering, unsupervised methods, parameter, word sense discrimination, context.

Introduction

In natural language processing word sense disambiguation is the task of automatic assignment of a correct sense from a predetermined sense inventory to a polysemous word. It is tightly related to the task of machine translation, where a correct translation of a word or phrase must be chosen from a list of translation candidates. Recently, the task of selection of the best translation or several interchangeable (synonymous) translations for a given source word in context and a set of target candidates has become known as word translation disambiguation.

All these tasks require contextual information to resolve an ambiguity, albeit translational or semantic.

In natural language processing approaches that involve a manually tagged training corpus that is further used for training of a machine-learning algorithm are known as supervised methods. Methods that automatically learn from "raw" corpus are called unsupervised. There are also approaches that are based on manually crafted rules or use existing dictionaries or heuristics that are known according to [1] can be described as knowledge-based approaches.

In the past decade approaches to bootstrap machine translation with preliminary word sense disambiguation or word sense translation were explored in [23, 2-5]. These approaches are based on supervised WSD classifiers that require extensive training on a large manually tagged training corpus. They are resource-demanding and provide relatively little or no improvement at a high cost.

The task of word translation disambiguation was treated independently in [8, 12] either with a supervised classifier or with huge annotated monolingual corpora.

As it was noted in various reviews [1, 13], supervised methods have achieved substantial results but they require very costly training corpora, which is normally tagged by human experts. The training corpora have become a bottleneck of this approach and since its results anyway do not reach a human-made gold standard [7], ultimately the attention of researches has been driven to unsupervised methods.

There are two main directions in unsupervised methods: methods that use monolingual corpora and look for similarities in contexts or documents, as in context or document clustering, and methods that extract information from word aligned multilingual corpora also known as cross-lingual methods.

Context clustering is an unsupervised approach to detection of similarities in contexts [16, 20]. Its results highly depend on parameters used for clustering. This approach was applied to word sense discrimination [19], which is mere distinguishing between different senses.

Diab and Resnik [6] use cross-lingual approach for unsupervised word sense tagging. The authors use a word-aligned French-English parallel corpus with a tagged part in English to tag its French part with

corresponding senses. This approach is aimed to facilitate sense-tagging of other languages given a broadly sense-tagged corpus in English. Consequently, although the suggested method is unsupervised, it requires substantially tagged data.

As follows from the above, the suitability of unsupervised approaches to word translation has not been explored. Our hypothesis is that unsupervised context clustering along with word aligned parallel texts can serve for obtaining context characteristics that would allow correct selection of a translation candidate for a word in a context in unsupervised manner. In this work we explore the suitability of context clustering for word translation disambiguation by comparing clustering results for various parameter combinations and evaluating them against known translation classes. In particular, we explore several parameter combinations with values that were found to be the best for the tasks of document and context clustering in [21, 24, 19, 11, 15]. For evaluation of clustering results we use translation equivalents that were obtained from word aligned parallel corpus.

The paper is organized as follows. In Section 2 we give a short overview of the parameters involved in context clustering. Section 3 describes experimental settings including context clustering software, dataset and the procedure for detection of dataset classes used for evaluation and interpretation. In Section 4 we demonstrate the obtained results and perform their analysis. Section 5 provides conclusion remarks and outlines future work in this direction.

2 Context Clustering

In the past decade the topic of unsupervised word sense discrimination, that is discrimination between different word usages *in context*, was actively investigated [1, 13]. The most known solution to this problem is clustering of contexts that contain a word in question, which is a particular application of document clustering [16, 20]. A great review of clustering as unsupervised classification of dataset elements into groups is provided in [9]. The clustering algorithms that are suitable for document clustering are described and analyzed in [21] and implemented first in CLUTO clustering toolkit [10], which receives an extension in SenseClusters [18].

However, results of context clustering highly depend on a variety of parameters: clustering algorithms, criterion functions for cluster detection, context representations, context similarity measures, and cluster stopping criteria.

2.1 Features

To perform a clustering one has to choose features that would represent each element of a dataset. In the field of document and context clustering each element, i.e. a document or a context, can be represented as a vector in a feature space. For example, a document can be represented as a vector of term frequencies:

$$dtf=(tf_1, tf_2, \ldots, tf_n),$$

where tf_i is the freqryrcy of a particular term *i* in a document and *i* through *n* are all terms from the entire document set. Naturally, a document cannot contain all terms. Therefore, many of the dimensions of a term vector will be equal to zero.

Features are called *unigrams*, when only one-word terms are considered. Pairs of two consecutive words are called *bigrams*. Yet in this work we adopt the extension of bigram's definition that is introduced in [17] and implemented in SenseClusters [18]. The extended definition states that bigrams are pairs of words that occur in a given order within some distance from each other. The distance is called *window*. For example, for a window of size five there could be at most three intervening words between the first and the second word that make up a bigram. On contrast to the bigrams, unordered pairs of words within a given window are called *co-occurrences*. These three types of features are suitable as for "headless" contexts or documents when there is no target word, the senses of which one wants to discriminate, as well as for "headed" contexts that contain a marked word. For the latter a feature called *target co-occurrence* is introduced. Target co-occurrences are co-occurrences that include the marked word.

Naturally, there are words such as auxiliary verbs, articles, conjunctions, etc., that are common for any context and, therefore, do not bring in any characteristic information. Such words are known as stopwords and are not considered as features.

On the other hand, words that occur very seldom to be a solid basis for context grouping must be excluded from the feature list as well. In SenseClusters frequency-cut parameter *r* serves as a threshold to exclude features that occur less than *r* times.

2.2 Order of context representation

First-order and *second-order representations* for short contexts are analyzed in [15]. The first-order representation represents a context as a vector only of those features that directly present in the context. The second-order representation also considers features that co-occur with the initial context features in other context. For example, if we have context 1 *"computer mouse"* and context 2 *"wireless mouse"*, "wireless" is a second-order feature for context 1 and "computer" is a second-order feature for context 2. Pedersen [15] shows the second-order representation to be better for short contexts since they contain fewer words than a document. Both representations are implemented in SenseClusters toolkit.

2.3 Similarity measure

To evaluate similarity between contexts, a *similarity measure* must be introduced on the selected feature representation. If elements are represented as feature vectors, such similarity measures as distance or cosine can be used. In document clustering similarity the most commonly used measure is the cosine:

$$cosine(\boldsymbol{d1}, \boldsymbol{d2}) = (\boldsymbol{d1} \bullet \boldsymbol{d2}) / \|\boldsymbol{d1}\| \|\boldsymbol{d2}\|.$$

Hence, contexts can be either represented in a *vector space*, where a vector corresponds to each document, or a *similarity matrix* can be constructed based on pairwise similarities between contexts.

2.4 Clustering criterion functions

The task of clustering is optimization of a clustering criterion function, which is a function from similarity measure. Cluster criterion functions can be internal or external. Internal criterion functions take into account only elements of a particular cluster and do not consider elements from other clusters. On the contrast, external criterion functions focuses on (dis)similarity between clusters. A review and comparison of criterion functions for partitional clustering is presented by [24]. The authors evaluate the performance of seven different criterion functions for the problem of document clustering. They show that two of the seven criterion functions (I_2 and H_2) steadily provide good results with most of the clustering algorithms, while some of the rest give better results under specific conditions on element density in a cluster, e.g. *UPGMA*, which, strictly speaking, is rather a cluster similarity measure designed for agglomerative clustering.

2.5 Clustering techniques

A variety of clustering techniques and algorithms exists to determine the sequence of steps for grouping and further regrouping of elements. This variety can be classified into three groups: hierarchical clustering, partitional clustering, and hybrid.

The most popular algorithm of the hierarchical clustering is agglomerative clustering, which first considers each element as a separate cluster and then groups them. Hierarchical clustering that works in the opposite direction is called divisive. The basic algorithm for the agglomerative clustering is:

1. Compute the similarity between all pairs of clusters.
2. Merge the most similar (closest) two clusters.
3. Update pairwise similarities between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

Patitional clustering divides a whole dataset into a given number N of clusters at once by randomly selecting N initial points as cluster centroids and then optimizes the clustering solution by reorganizing the elements. Hence, it does not have a hierarchy of clusters but rather a "flat" solution. Here we present the algorithm known as basic k-means clustering:

1. Select K points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

A comparison of clustering techniques concerning document clustering is performed by Steinbach *et al*. [21]. As it follows from their research, the best clustering techniques for document clustering were bisecting k-means among flat clustering techniques and refined agglomerative clustering with UPGMA similarity function among hierarchical techniques. They also showed that bisecting k-means technique performed better than refined agglomerative clustering with UPGMA.

2.6 Cluster stopping measures

However, the existing clustering techniques imply that a number of clusters is already known, which is not true in many cases, especially for document and context clustering. A solution for automatic cluster stopping was suggested in [11] along with four cluster stopping measures: *gap*, which is based on gap statistics, *pk1*, *pk2*, and *pk3*.

2.7 Clustering evaluation

Additionally, evaluation of context clustering results is not an easy task. There are many different quality measures and the performance ranking of a clustering algorithm depends substantially on which measure is used [21].

Two basic classes of clustering quality measures exist. Internal quality measures do not use any external knowledge and are based on similarity or dissimilarity functions used to form a clustering solution. Hence, the result of such evaluation directly depends on the clustering function used and such evaluation is difficult to interpret for a set of contexts from the point of view on their information content. External quality measures compare clustering results to known classes of dataset elements. Here the problem is to obtain these classes.

In Section 3 we describe how we obtain the classes for the dataset. Consequently, we use external quality measures of entropy and purity to evaluate the results.

Given a particular cluster S_r of size n_r, the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{n_r^i}{n_r}\log\frac{n_r^i}{n_r},$$

where q is the number of classes in the dataset, and n^i_r is the number of elements of the i^{th} class that were assigned to the r^{th} cluster. The resulting entropy is the weighted sum of all entropies:

$$E = \sum_{r=1}^{k}\frac{n_r}{n}E(S_r),$$

where k is a number of clusters and n is a number of all elements in the dataset.

Entropy looks at how various classes of dataset elements are distributed between clusters obtained for the same dataset. In brief, the lower is the entropy, the better.

The purity of a cluster is calculated as:

$$P(S_r) = \frac{1}{n_r}\max_i n_r^i,$$

It is the largest fraction of a cluster that is formed by the elements of the same class or in other words a fraction of the largest class in a cluster.

The overall purity of a clustering solution is a weighted sum of purities of all individual clusters:

$$P = \sum_{r=1}^{k}\frac{n_r}{n}P(S_r),$$

The case when each cluster is formed by the elements of one class is the best and gives the highest purity.

3 Experimental Settings

Further we describe experimental settings used at the experiments.

3.1 Clustering parameters

In this work we perform context clustering with SenseClusters toolkit [18]. It is a complete and freely available context clustering system that provides support for feature selection from large corpora, several different context representation schemes, various clustering algorithms, and evaluation of the discovered clusters.

Parameters with fixed values We set values of several parameters to be unchangeable and regarded as "default" for our experiments:

- the order of feature representation is set to *-o2*, which stands for the second order;
- the context are represented as feature vectors in *vector space*;
- window is set to 5;
- frequency-cut parameter *r* is set to 3.

We chose the second-order context representation since it is shown to be better for short contexts (Pedersen, 2008).

The vector space is preferred over the similarity matrix representation based on the work of Purandare and Pedersen [19]. They analyzed 6 combinations of 4 clustering parameters: order of context representation, features, vector space/similarity matrix, and clustering method. Purandare and Pedersen show that the best results were achieved for combinations with vector space.

For the value of window parameter we took as a reference the work by Purandare [17], where this parameter was set to 5.

The value of the frequency-cut parameter is chosen heuristically based on the assumption that for the size of our dataset, which contains 1449 contexts, a higher number might cut out significant features, whereas a lower number would be unreasonable.

Parameters with varied values In the experiment we varied several parameters: features for context representation, clustering methods, criterion functions and cluster stopping criteria. Since the total number of possible combinations is very high, we analyzed only among those parameter values that are proved to be the best for document and context clustering in [21, 24]. We also considered repeated bisections and refined repeated bisections methods since they are considered in works on context clustering [19, 20]. The parameters with their varied values are:

- features for context representation: unigrams, bigrams, co-occurrences, target co-occurrences;
- clustering methods: direct k-means, repeated bisection, refined repeated bisection, agglomerative;
- criterion functions: I_2 and H_2 for partitional methods and UPGMA for the agglomerative method;
- cluster stopping measures: gap, pk1, pk2, pk3.

The total number of experiments is 112.

3.2 Dataset

We used sentence aligned English-Spanish Europarl parallel corpus from OPUS open corpus [22] to extract contexts for clustering and to detect translation equivalents.

For our purpose of exploring context clustering suitability for word translation disambiguation, an ambiguous word had to satisfy the following criteria:

- to have a number of instances in a chosen parallel corpus that would be sufficient for unsupervised clustering (we set it 1000);
- to have more than one candidate translation in the parallel part of a corpus.

We considered as candidates only those translations, the number of entries of which was at least 1-2% of the source word instance number.

The analysis of the above criteria was performed using OPUS word alignment database. We have chosen

several words that satisfy these criteria. Due to time constraints, we present results only for the word "FACILITY".

As a context we used an extract of seven consecutive sentences from the corpus, a sentence with the chosen source word being the forth. At this step we extracted 1771 contexts for our dataset.

The dataset was converted to lower-case and tokenized.

To evaluate clustering results we needed to detect corresponding translations. First, we performed word alignment automatically with GIZA+ + [14]. Yet we obtained excessively many word-to-NULL alignments. It might be due to a relatively small size of the dataset corpus, which additionally contained nearly 20% of wrong sentence alignments.

Therefore, we developed an alternative approach to detection of corresponding translations for a selected source word. A detailed description of the procedure and results are described in [25]. For *ca.* 600 entries of our dataset, pruned alignments were available from OPUS word alignment database. The rest was detected manually by comparing source word contexts with their corresponding parallel contexts.

At this stage we detected 342 contexts (~ 20% of the dataset) that were wrongly sentence-aligned as the example in Table 1 shows:

I hope that as soon as possible we will have the financial perspective and the Stability Instrument, which should, under normal circumstances, finance the Peace <head>Facility</head> and enable the problem to be resolved.	Comparto, pues, este punto de vista.

Table 1

We deleted such contexts from our dataset.

There were also cases when the word "facility" did not have a direct translation equivalent as in Table 2:

… but the lack of financing and credit <head>facilities</head>	… pero la falta de financiación y créditos.

Table 2

We tagged such cases as NOTAG since we wanted to detect whether unsupervised sense clustering would find something in common between contexts that are translated in this manner.

Further, there were about 100 translations that had a very low frequency of 1 to 6 and could not be considered independent translation candidates. We performed manual grouping of them with their synonyms based on the contexts where they appeared. There were 10 instances that we decided to mark as NOTAG since the translator's decision to choose a particular translation equivalent was not clear and easily deducted.

In the end, we obtained a dataset of 1429 contexts with 21 translation classes including NOTAG. The dataset along with a translation candidate key file and information on some intermediate steps can be found at *www.gelbukh.com/resources/word-translation-alignments*.

According to monolingual dictionaries we consulted (Online Merriam-Webster, Oxford Concise Thesaurus, WordNet, and Larousse American Pocket), they distinguish between 4 and 5 senses for the word "facility" that can be described as:

- installation, building;
- service;
- equipment;
- possibility;
- readiness.

We took these numbers as guidance for the minimum number of clusters. Therefore, any combination of parameter values that gave fewer than 4 clusters was discarded from the comparison of parameter values.

4 Experimental results

The number of clusters that we obtained with various clustering parameter combinations varied from 1 to 6.

Table 3 shows the frequencies of each number of clusters for a cluster stopping measure.

cl. num.	**1**	**2**	**3**	**4**	**5**	**6**
gap	24	0	4	0	0	0
pk1	11	10	3	1	3	0
pk2	0	8	10	3	4	3
pk3	0	12	9	6	1	0

Table 3

As it follows from Table 3, cluster stopping measures *gap* and *pk1* provide the lowest number of clusters. Gap statistic measure gives no results that would be higher than the threshold of 4 clusters. Pk1 measure gives acceptable results only in 4 cases, which is 3.5% of all cases.

The fractions of experiments for each cluster number from the total number of experiments are shown in Table 4.

cl. num.	**1**	**2**	**3**	**4**	**5**	**6**
fraction, %	31.2	26.8	23.2	9.0	7.1	2.7

Table 4

Of the total number of experiments 50% were for cluster numbers 2 and 3, and only 18.8% (21 of a total of 112 combinations) passed the threshold of 4 clusters.

An assumption that the word "facility" may have only 2 to 3 "*real"* or well distinguishable senses does not seem to be probable. If we take a look at the list of generalized senses for "facility" in Section 3, they hardly can be grouped into a number of independent and non-intersecting senses less than four. And if we take into account that a lexical company of a word in context might vary even more than its semantic meaning, we would rather expect a larger number of clusters than a smaller one.

Therefore, we interpret the steadily low number of clusters for cluster stopping criteria gap and pk1 as a quality of these criteria. *Pk2* and *pk3* measures give acceptable results in 36% and 25% of their usage cases respectively.

The parameter values, the entropy, and the purity for cases with the cluster number more than 4 are presented in Table 5.

clmeth	**crfun**	**clstop**	**cl #**	**E**	**P**	**clmeth**	**crfun**	**clstop**	**cl #**	**E**	**P**
Co-occurrences						**Unigrams**					
agglo	upgma	pk2	6	80.6	25.5	agglo	upgma	pk2	6	84.1	24.2
direct	h2	pk1	4	80.4	25.6	direct	i2	pk2	6	**74.8**	26.9
direct	h2	pk3	4	80.4	25.6	rb	h2	pk1	5	75.2	**28.3**
direct	i2	pk2	5	80.2	25.5	rb	h2	pk2	4	76.2	27.6
direct	i2	pk3	4	80.4	25.6	rb	h2	pk3	4	76.2	27.6
rb	h2	pk1	5	80.7	25.0	rb	i2	pk2	5	75.6	27.8
rb	h2	pk2	4	81.0	25.0	rbr	h2	pk1	5	75.2	**28.3**
rb	h2	pk3	4	81.0	25.0	rbr	h2	pk2	4	76.2	27.6
rb	i2	pk3	5	80.7	25.0	rbr	h2	pk3	4	76.2	27.6
rbr	h2	pk3	4	80.4	25.6	rbr	i2	pk2	5	75.3	**28.3**
rbr	i2	pk2	5	80.2	25.5						

Table 5

As one can observe, no combinations with bigrams that are two consecutive words or target co-occurrences that are co-occurrences with the target word "facility" passed the threshold. For parameter combinations containing these features the number of clusters was lower than 4. This fact might be explained that conditions imposed on these features are hard to satisfy: bigrams require repeated consecutiveness of a word pair and target co-occurrences require co-occurrence with a target word within a certain window. On the one hand, a larger window size for these features might bring in more significant features. On the other hand, a

window of size more than 5 will introduce too much noise.

Several parameter combinations with *unigram* and *co-occurrence* features passed the threshold. It can be observed from Table 5 that for partitional clustering techniques –direct k-means, repeated bisections and refined repeated bisections– variation of entropy and purity has some dependency on the number of clusters. For fixed number of clusters and context feature pairs of entropy and purity can be grouped into as few as one or two groups of equal values. For example, if we set a context feature to be co-occurrence and a number for clusters to be 4, in 4 of 6 cases (entropy; purity) = (80.4; 25.6) and in the rest of the cases (entropy; purity) = (81.0; 25.0). To detect the actual dependence further experiments are needed.

The best entropy and purity values correspond to the parameter combinations with unigram features. In general, the entropy for unigrams is about 5% better than the entropy for co-occurrences and the purity is 12% better for unigrams than for co-occurrences. Yet comparison of these entropy and purity values to those obtained in [21, 24] is hindered by the dependence of entropy and purity on the number of classes.

Our consideration is that the entropy and purity measures as they are described in Section 2.7 might be inappropriate for cluster evaluation in our task. These measures were intent to evaluate word sense discrimination results, when it is assumed that each cluster corresponds to a sense and it is expected (or manually set) that the number of clusters would be more or less the same as the number of senses. On the contrast, in our case it is completely acceptable if more than one class are clustered together, which corresponds to the case of synonymous translations, or if elements of one class are distributed between several clusters, which is the case of preserved homonymy.

To check how cluster number will influence the entropy and purity, we performed an experiment with the number of clusters manually set to 21, which is the number of our translation classes. In this experiment we used a clustering parameter combination that gave the highest purity. The results are shown in Table 6.

clmeth	**crfun**	**clstop**	**cl #**	**E**	**P**
Unigrams with fixed number of clusters					
rb	h2	n/a	21	67.2	32.7

Table 6

As it can be seen, the more than 4 times increase of the cluster number from 5 to 21 improves the values of entropy and purity only 10.6% and 15.5 % respectively.

5 Conclusions and future work

In this work we perform comparison of various clustering parameter combinations and explored suitability of context clustering application to unsupervised word translation.

The number of clusters more than the threshold of 4 occurred only for 18.8% of the experiments. Numbers of 2 and 3 were detected in 50% of cases. Yet these results cannot be interpreted from the semantic point of view, therefore, they were discarded as it was initially intended. However, formal analysis of semantic similarity of senses through an ontology or semantic hierarchy can give new perspective on these numbers.

We detected that cluster stopping measures gap and pk1 provide very low numbers of clusters that cannot be interpret from the semantic point of view. The numbers of clusters that correspond to the semantic assumption of the number of word senses can be achieved in most cases with pk2 and pk3 cluster stopping measure. Also pk1 cluster stopping measure should not be completely discarded since it provided 19% of all acceptable results.

We were not able to detect acceptable results for bigram and target co-occurrence features. It might be explained by inappropriate window size and data sparseness that in our experiments was not handled through singular value decomposition. Hence, further experiments with singular value decomposition and varying window size are necessary.

The evaluation of results through entropy and purity gives us the numbers that are not easily interpreted in the task of word translation when the number of classes is much higher than the number of clusters. Hence, we will work on development of different quality measure that would be more adequate for our goals.

References

1. Agirre E., Edmonds P. (eds.) (2006), *Word Sense Disambiguation. Algorithms and Applications*, Springer.
2. Carpuat M., Wu D., Word sense disambiguation vs. statistical machine translation. *Proc. of the annual meeting of the ACL*, 2005, pp. 387–394.
3. Carpuat M., Wu D., Evaluating the word sense disambiguation performance of statistical machine translation. *Proc. of the Second International Joint Conference on Natural Language Processing (IJCNLP)*, 2005, pp.122–127.
4. Carpuat M., Wu D., Improving statistical machine translation using word sense disambiguation, *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007),* 2007, pp. 61–72.
5. Chan Y.S., Ng H.T., Word sense disambiguation improves statistical machine translation, *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007, pp. 33–40.
6. Diab M., Resnik P., An unsupervised method for word sense tagging using parallel corpora, *Proc. of the 40th Annual Meeting on Association for Computational Linguistics,* 2002, pp. 255-262.
7. Gale W., Church K.W., David Yarowsky D. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. *Proc. of the 30th Annual Meeting of the Association for Computational Linguistics,* Newark, Delaware, 1992.
8. Holmqvist M., Memory-based learning of word translation, *Proc. of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007,* Tartu, Estonia, 2007, pp. 231-234.
9. Jain A.K., Murty M.N., Patrick J. Flynn P.J. (1999), Data Clustering: A Review. *ACM Computing Syrveys*, vol. 21, pp. 264-323.
10. Karypis, G. (2003), CLUTO - A Clustreing Toolkit, *University of Minnesota, Department of Computer Science Technical Report 02-017*.
11. Kulkarni A., Pedersen, T. (2006), Unsupervised Context Discrimination and Automatic Cluster Stopping, MS Thesis, *University of Minnesota Supercomputing Institute Research Report UMSI 2006/90*.
12. Marsi E., Lynum A., Bungum L., Gambäck B., Word Translation Disambiguation without Parallel Texts. *Proc. International Workshop on Using Linguistic Information for Hybrid Machine Translation*, Barcelona, Spain, 2011.
13. Navigli R. (2009), Word sense disambiguation: A survey, *ACM Computing Surveys,* vol. 41(2), pp. 1-69.
14. Och F.J., Ney H. (2003), A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, vol. 29(1), pp. 19-51.
15. Pedersen T. (2008), Computational Approaches to Measuring the Similarity of Short Contexts: A Review of Applications and Methods, *University of Minnesota Supercomputing Institute Research Report UMSI 2010/118*.
16. Pedersen T., Bruce R., Distinguishing word senses in untagged text, *Proc. of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, 1997, pp. 197–207.
17. Purandare A. (2004), Unsupervised Word Sense Discrimination By Clustering Similar Contexts. MS Thesis. University of Minnesota.
18. Purandare A., Pedersen T., SenseClusters - Finding Clusters that Represent Word Senses. *Proc. of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04),* 2004, pp. 26-29.
19. Purandare A., Pedersen T., Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, 2004, pp. 41-48.

20. Schütze, H. (1998), Automatic Word Sense Discrimination. *Journal of Computational Linguistics*, vol. 24(1), pp. 97-123.

21. Steinbach M., Karypis G., Kumar V. (2000), A comparison of document clustering techniques, *University of Minnesota, Technical Report 00-034*.

22. Tiedemann J. (2009), News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces, *Recent Advances in Natural Language Processing*, vol. V, pp. 237-248.

23. Vickrey D., Biewald L., Teyssier M., Koller D., Word-sense disambiguation for machine translation. *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing 2005*, 2005, pp. 771–778.

24. Zhao Y., Karypis G. (2001), Criterion Functions for Document Clustering: Experiments and Analysis, *University of Minnesota, Department of Computer Science Technical Report 01-040*.

25. Zhila A., Gelbukh A. (2012), Gold standard word-aligned corpus for selected terms. Submitted.