Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 25–32

ISBN 978-83-60810-27-9
ISSN 1896-7094

Evaluation of Clustering Algorithms for Polish Word Sense Disambiguation

Bartosz Broda, Wojciech Mazur
Institute of Informatics, Wrocław University of Technology, Poland
bartosz.broda@pwr.wroc.pl, wojciech.mzr@gmail.com

***Abstract*—Word Sense Disambiguation in text is still a difficult problem as the best supervised methods require laborious and costly manual preparation of training data. Thus, this work focuses on evaluation of a few selected clustering algorithms in task of Word Sense Disambiguation for Polish. We tested 6 clustering algorithms (K-Means, K-Medoids, hierarchical agglomerative clustering, hierarchical divisive clustering, Growing Hierarchical Self Organising Maps, graph-partitioning based clustering) and five weighting schemes. For agglomerative and divisive algorithm 13 criterion function were tested. The achieved results are interesting, because best clustering algorithms are close in terms of cluster purity to precision of supervised clustering algorithm on the same dataset, using the same features.**

I. Introduction

WORD Sense Disambiguation (WSD) deals with contextual resolution of lexical ambiguity. Most words in natural language have more than one lexical meaning (sense), but usually only one of them is active in a given context. Typical example of ambiguous word is *line*, which according to WordNet (an electronic thesaurus, cf. [1]) has 36 senses. WSD is important problem for applications in domain of Natural Language Processing (NLP). Machine translation cannot work without some form of disambiguation, but WSD can be helpful also for information retrieval, information extraction and computer aided lexicography among others [2].

WSD is a hard problem. Most difficulties arise from the fact that the concept of a meaning is vague. Usually, there are no clear boundaries between one sense or the other [3]. Typically, the problem of defining meaning is tackled with using dictionaries (which are called *sense inventory* in a context of WSD). I.e., from the algorithmic point of view sense inventories are used to enumerate all the meanings that a given word has. Now, the goal of WSD can be stated as choosing appropriate sense from sense inventory in a given context of a word.

There are two main approaches to WSD based on machine learning: supervised and unsupervised [2].[1] Supervised learning focuses on the usage of manually disambiguated examples of text snippets containing ambiguous words. We need to choose an appropriate sense inventory in advance, at early stages of the construction of supervised WSD system. Some features are extracted from those text snippets (or contexts[2]) and classifiers are trained using this manually labeled data. Most of the time, supervised approaches are superior to unsupervised in terms of accuracy of automatic disambiguation when used on the same type of texts that the systems were trained on.

Nevertheless, there is another issue connected with the problem of the definition of a meaning, i.e., an issue of creation of other resources used for automatic system performing WSD. This is especially evident in creation of corpora[3] manually annotated (tagged) with senses, which are used for training machine learning classifiers in a supervised setting. There are two important problems during manual sense tagging of a corpus: low *interannotator agreement* (IA) and high cost of annotation process. IA is a way of measuring how much annotations assigned by one annotator differers from annotations assigned by another annotator. IA is used for estimation of an upper bound on performance on automatic WSD. Typically, it is not enough to give a value of percentage agreement, because agreements and disagreements may arise by chance. Cohen's κ is widely used in computational linguistic community for this purpose, but there are also other measures [5]. The cost of annotation is high, because large effort is required during manual annotation. Mihalcea estimated that a construction of a corpus with sufficient amount of data for supervised classification algorithms for 20 000 ambiguous words would require 80 man-years of work [6].

On the other hand, unsupervised and semi-supervised algorithms can be used. The amount of manual labor required is much lower in learning without supervision. Unsupervised approaches to WSD tend to use unlabeled data and automatically find sense distinctions. Usually those methods involve some form of clustering. Harris' distributional hypothesis [7] can be used as a theoretical foundation for unsupervised methods of WSD. It states that "meaning of entities (...) is related to the restrictions on combinations of these entities relative to other entities.". In this context entities can be understood as words.

The main goal of this work is to compare various clustering algorithms in the task of unsupervised Word Sense Disambiguation for Polish data. In unsupervised WSD system deals with grouping of contexts for given word that express the

[1]There is a plethora of other approaches to WSD, e.g., based on translational equivalence or hand-written rules. We omit those for brevity. For extensive overview of other methods see, e.g., [2], [4].

[2]We will use term *context* to denote a passage of text containing ambiguous word.

[3]Here we define a corpus as a collection of texts prepared for linguistic processing

same meaning without providing explicit sense labels for each group (e.g., without using a dictionary) [8]. Also, this work is motivated by the fact that clustering is important for semi-supervised WSD algorithm called Lexicographer Controlled Semi-automatic word Sense Disambiguation [9], [10]. So far, the selection of the algorithm used in LexCSD was motivated by the performance of the given algorithm in other tasks and its analytical properties, because analysis of the performance of different clustering algorithms in similar settings (i.e., using similar dataset and features) for Polish WSD is difficult to find.

There are a few differences when dealing with WSD data in comparison to classical applications of clustering. To name just a few: the distributions of classes (senses) are skewed[4], data is represented in spaces of very large number of dimensions (thousands or even hundreds of thousands), for some classes only very specific, often overlapping among classes features are important and sometimes there is difficulty in distinguishing between two close classes.

The paper is organized as follows. First the selected clustering algorithms are briefly described. Evaluation section starts with the analysis of evaluation metrics used. Next, the corpus and experimental settings are described. Section III-D provides discussion of results. Section IV gives a summary of performed experiments and overviews direction of further works.

II. Selected algorithms for testing

For this work we have selected a few classical clustering algorithms, but we tried to choose algorithms representing a few different approaches to the problem of clustering. We started with *K-means* and *K-medoids* algorithms, which represent simple, hard and flat clustering methods. We choose *Growing Hierarchical Self-Organising Map* (GHSOM) as a representative of family of clustering using neural networks. GHSOM is also a hierarchical clustering algorithm. We experiment with standard hierarchical clustering algorithms with different criterion functions, both from agglomerative and divisive families of algorithms. Last but not least, we test also graph-based clustering algorithm. We have reimplemented K-means, K-medoids and GHSOM and use existing implementation of other algorithms [11].

We are focusing on clustering for WSD so we will use NLP-related terminology during description of algorithms. As a task of WSD is a contextual one, we will cluster *contexts* (text snippets) containing ambiguous word. From the context some real-valued features are extracted. So the context is a vector of features $\vec{v}$ in high dimensional space. We will use term *context* and *context vector* interchangeably. The exact nature of context and feature extraction process are described in Sec. III-B.

A. K-means and K-medoids

K-means is one of the simplest clustering algorithm. K-means defines cluster as a centers of mass of contexts being clustered [12]. Those centres are represented as centroids. Initially random contexts are chosen as centroids. Then we assign most similar contexts to each centroid. After this step new centroids are computed as a mean of all the contexts in a group. This process is then repeated until some stopping criterion is reached, e.g., number of iteration reaches some predefined threshold or the clustering solution do not change significantly between subsequent iterations.

K-medoids is similar in concept to K-means algorithm. The most fundamental difference between the two algorithms is that K-medoids uses real contexts from the dataset as a basis for clustering in contrast to centroids used in K-means (which are artificial contexts). One of the realisations of K-medoids is an approach called *Partition Around Medoids*, or PAM [13]. In PAM one starts with randomly selection of initial medoids. Then every swapping of every medoid with every context is tested in terms of decreasing *cost* of whole clustering solution. This approach has its drawbacks in terms of computational complexity, i.e., $O(k(n-k)^2)$, where n is number of contexts to cluster and k is number of medoids. Thus a few extensions have been proposed that, e.g., employ sampling (CLARA) or randomized search (CLARANS) [13]. Nevertheless, we use classical PAM, as both mentioned algorithms can have negative impact on quality in comparison to PAM. This approach is applicable in our experiments, as we use relatively small datasets.

B. Growing Hierarchical Self-Organizing Map

The Growing Hierarchical Self-Organizing Map (GHSOM) [14] is a natural extension of Kohonen's idea of Self-Organizing Maps (SOM) [15]. SOM is an *artificial neural network* consisting of many neurons. Every neuron consists of a weight vector. Training SOM is done in an unsupervised manner applying *winner takes most* strategy. Every feature vector is delivered to the network input several times. For every input vector the similarity with the neuron weight vector is computed. Weights of the most similar neuron (the winner) and its neighbourhood are updated to be even more similar to the input pattern. The learning algorithm is constructed in such a way, that the neighbourhood and the degree of the weight updating is decreasing over time.

GHSOM address one of the most important drawback of SOM — the a priori definition of the map structure. Rauber *et al.* proposed an algorithm for growing SOM both in a terms of the number of map neurons and the hierarchy [14]. After the training stage of SOM mean quantization error for every neuron i (mqe_i) is calculated as the average distance of every context recognised by the neuron i to its weight vector. The average MQE_j for whole map on level j is computed, too. If $MQE_j \geq \tau_1 \cdot MQE_{j-1}$ then the additional row or column of neurons is added to the map and the training stage is repeated. In the other case the mqe_i for every neuron is compared to MQE_j. If $meq_i \geq \tau_2 \cdot MQE_{j-1}$ then another layer of the map is created for contexts recognised by the neuron i.

[4]Not all senses are represented in the data equally; distribution of senses is biased towards a few frequent senses.

C. Agglomerative and Divisive Clustering

Agglomerative and divisive clustering algorithms produce *hierarchical* clustering trees called dendrograms. Agglomerative clustering starts in a situation that each context is contained in a separate cluster, then in each step two clusters maximising *criterion function* are merged. On the other hand, divisive algorithms starts with all contexts in one cluster which are repeatedly bisected according to the criterion function. We are using existing implementation of hierarchical algorithms from CLUTO[5] [11]. We use *rbr* variant of divisive algorithm, i.e., standard bisecting clustering is employed and is further optimized according to criterion function [16].

Criterion function is very important aspect of both agglomerative and divisive clustering algorithms as it drives the whole process. There are many criterion function available [17]. We have tested standard criterion functions used with agglomerative algorithms, i.e.: single link (slink), complete link (clink), average link (upgma) and weighted variants of single (wslink), complete (wclink) and average links (wupgma).

The second group of criterion function including $i_1, i_2, \varepsilon_1, G_1, G'_1, H_1, H_2$ can be used with both agglomerative and divisive algorithms. The exact form of those functions are given by [11]:

$$I_1 = \text{maximize} \sum_{i=1}^{k} \frac{1}{n_i} \Big(\sum_{\vec{v},\vec{u} \in S_i} sim(\vec{v},\vec{u}) \Big) \quad (1)$$

$$I_2 = \text{maximize} \sum_{i=1}^{k} \sqrt{\sum_{\vec{v},\vec{u} \in S_i} sim(\vec{v},\vec{u})} \quad (2)$$

$$\varepsilon_1 = \text{minimize} \sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} sim(\vec{v},\vec{u})}{\sqrt{\sum_{v,u \in S_i} sim(\vec{v},\vec{u})}} \quad (3)$$

$$G_1 = \text{minimize} \sum_{i=1}^{k} \frac{\sum_{v \in S_i, u \in S} sim(\vec{v},\vec{u})}{\sqrt{\sum_{v,u \in S_i} sim(\vec{v},\vec{u})}} \quad (4)$$

$$G'_1 = \text{minimize} \sum_{i=1}^{k} n_i^2 \frac{\sum_{v \in S_i, u \in S} sim(\vec{v},\vec{u})}{\sqrt{\sum_{v,u \in S_i} sim(\vec{v},\vec{u})}} \quad (5)$$

$$H_1 = \text{maximize} \frac{I_1}{\varepsilon_1} \quad (6)$$

$$H_2 = \text{maximize} \frac{I_2}{\varepsilon_1}, \quad (7)$$

where k is total number of clusters, S is total number of contexts to cluster, S_i is a set of contexts assigned to i-th cluster, $n_i = |S_i|$, and $sim(\vec{v},\vec{u})$ is similarity between two context vectors $\vec{v}$ and $\vec{u}$.

D. Graph Partitioning Based Clustering

We use an implementation of min cut graph partitioning algorithm from CLUTO [11]. This algorithm starts with creation of neighbourhood graph based on similarities between contexts and then applies min cut to partition the graph into disjoint regions. Min cut uses approach that the size of graph edges in a partition is minimal.

This approach achieved high quality in research on semi-automatic extension of Polish WordNet [18] and was also used in Polish WSD based on weakly-supervised settings using LexCSD algorithm [10].

III. Experiments

A. Evaluation Measures

Evaluation of clustering algorithms can be done in many ways [19]. Some of them are based on *external criteria*, i.e., the comparison of the resulting clustering solution with some pre-existing categories that were created manually. On the other hand, one can use an *internal criteria* without resorting to gold standard clustering. The most important drawback of evaluation using internal criteria is that good score does not always corresponds to good results of clustering in a given application [20]. As we have developed semantically annotated corpus (SCWSD, see Sec. III-B) we can use it for the need of evaluation. The problem with SCWSD is its small size, so there is a risk of not capturing all of the peculiarities and biases of some large corpora in SCWSD.[6]

We used several measures for evaluation to capture different aspects of created groups. For measuring how homogeneous clusters are we used *Purity*:

$$Purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j|, \quad (8)$$

where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$ is a set of clusters, a $C = \{c_1, c_2, \ldots, c_j\}$ — a set of pre-existing categories. In our setting C is a set of contexts with ambiguous word annotated with the same sense. $Purity(\Omega, C) \in \langle 0, 1 \rangle$, where 1 is the best case. A drawback of $Purity$ is its preference for solutions with large number of groups. Assigning every context to a singelton cluster gives Purity of 1 [20].

The *Rand Index* measures accuracy on the basis of decisions performed for the subsequent context pairs. If we use TP for *true positive*, TN for *true negative*, FN for *false negative* and FP for *false positive.* the Rand Index is given by the following equation:

$$R_I = \frac{TP + TN}{TP + FP + FN + TN} \quad (9)$$

One of the drawbacks of using R_I for evaluation is the equal treatment of false positives and negatives. Using decision for context pairs we can also use standard measures of information retrieval, i.e., precision P, recall R and the harmonic mean of precision and recall F_β:

[5]CLUTO is a free software package implementing several clustering algorithms including partitioning, agglomerative and graph-based. Available at: http://glaros.dtc.umn.edu/gkhome/views/cluto/

[6]On the other hand, the total size of the dataset, i.e., 1344 contexts (Tab. I), is not very small in comparison to, e.g., [16], where the smallest dataset has 878 elements and the largest — 4069 elements.

$$P = \frac{TP}{TP+FP} \quad (10)$$

$$R = \frac{TP}{TP+FN} \quad (11)$$

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R} \quad (12)$$

Another way of measuring clustering quality is to use *Normalized Mutual Information* (NMI). NMI takes into account the trade off between quality and number of clusters as opposed to Purity.

$$NMI(\Omega,C) = \frac{MI(\Omega;C)}{[H(\Omega)+H(C)]/2} \quad (13)$$

$$MI(\Omega,C) = \sum_{k,j} P(\omega_k \cap c_j) log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \quad (14)$$

$$H(\Omega) = -\sum_k P(\omega_k) log P(\omega_k), \quad (15)$$

where mutual information $MI(\Omega,C)$ is normalized by entropy $H(\Omega)$ and probabilities can be counted using maximum likelihood estimation (MLE). The normalization by entropy is performed for penalizing clustering solutions with large number of clusters, as entropy tends to increase to maximum with number of clusters. Thanks to the normalization $NMI \in < 0,1 >$, where 0 corresponds to random clustering.

Last method used for evaluation is a variation of F-Measure used by Kulkarni and Pedersen in SenseCluster system [21]. Its main idea is contained in a sentence: "One sense—one cluster". It means that each cluster must have unique sense label. The label of each cluster is determined by most frequent class of cluster members, but one sense label cannot be assigned to multiple clusters. In particular, having more clusters than senses force us to treat members of unlabeled groups as unclustered. With that assumption in mind the standard measures of precision P_p and recall R_p are defined as:

$$P_p = Purity(\Omega', C) \quad (16)$$

Where Ω' is set of labeled clusters, C is set of classes defined above.

$$R_p = \frac{\#hits}{\#total\ instances} \quad (17)$$

Where *#hits* is number of elements with sense accordant to cluster sense label, *#total instances* - number of all elements including outliers.

Having P_p and R_p, we use F_β, with $\beta = 1$:

$$F(P)_1 = \frac{2P_pR_p}{P_p+R_p} \quad (18)$$

B. Corpus Description

For the need of evaluation we have used recently developed, manually disambiguated corpus called *Korpusik US*[7] (rough English translation: Small Corpus for WSD, henceforth SCWSD) [10].

The corpus consists of parts of IPI PAN Corpus [22]. Only 13 ambiguous words were annotated. The chosen words represent the variety of different problems for WSD; some of the senses have homonymous character, i.e., they represent separate homonyms of the same morphological base form. The sense inventory was based on extended version of Polish WordNet [18]. Performing evaluation only on limited set of words is called *Lexical Sample Task* [2]. The following words were chosen for annotation:

- agent: a person who represents a company or artist, secret agent, chemical agent
- automat: automaton, machine, telephone, a coin-operated automatic machine, submachine gun, automatic car transmission ;
- dziób: beak, bow, nose, front part of a ship, informal mouth, face (semantically marked)
- język: tongue, natural language
- klasa: category, class, rank, classroom, mathematical category, savoir-vivre, social class, subject, excellence
- linia: line, route, edge, line separating two areas, power line, assembly line, telephone line, row, lineage, contour, figure, ruler, line of defence, line of products for sale, credit line, geometric line.
- pole: field, area, playing field, physical field.
- policja: police (organization), police station, 'policemen'.
- powód: reason', plaintiff.
- sztuka: art, act of craftsmanship, item, a beautiful girl, dramatic play, theatrical performance of a play, an amount of fabric (for example wool), bale, a piece of meat.
- zamek: castle, lock, zipper, breechblock, trap in hockey, a part of machine or any device that stops its action.
- zbiór: set, group, mathematical set, collection, harvest, an act of harvesting, an exercise book, file.
- zespół: team, band, group of machines, complex of buildings, syndrome, sport team, botanical 'association'.

The annotation of the corpus was done by two native speakers of Polish: a professional linguist and a computational linguist. The exact corpus statistics are show in Tab. I. There are 1344 annotated examples. After the annotation process we measured interannotator agreement using Cohen's κ [23]. The agreement is surprisingly high, 0.88 for whole corpus. Such an agreement is very high in comparison with other corpora annotated with fine-grained WordNet-based senses [2].

SCWSD was previously used in research on WSD for Polish [10]. Its previous version (sense inventory base on early version Polish WordNet) was also used in [9], [24]. The best reported precision in [10] using supervised classifiers is 72.42%. There is also a baseline associated with a corpus called Most Frequent Sense baseline (MFS), i.e., using a heuristic classifier that chooses always the most frequent sense. For SCWSD the MFS baseline is 44.56% (weighted average for all the words).

[7]The corpus is available for browsing http://nlp.pwr.wroc.pl/webann

TABLE I
SMALL CORPUS FOR WSD (SCWSD) — STATISTICS

Word	No. of senses	Annotated senses	Examples	κ
agent	5	1/9/3/47/10	70	0.80
automat	5	1/24/30/4/46	105	0.97
dziób	4	28/13/31/9	81	0.98
język	3	3/23/49	75	0.97
klasa	11	15/6/12/11/14/31/10/8/1/10/1	119	0.80
linia	13	13/3/2/2/4/2/11/13/4/3/1/2/21	81	0.72
pole	5	1/1/23/25/46	96	0.86
policja	3	17/25/22	64	0.73
powód	2	136/122	258	0.98
sztuka	6	12/10/2/11/41/19	95	0.84
zamek	4	18/19/36/19	92	1.00
zbiór	5	32/7/8/31/9	87	0.87
zespół	6	10/4/28/58/1/20	121	0.95

C. Experimental Settings

The experimental settings are the same as in experiments presented in [10], where approaches based on supervised and weakly supervised were tested. We use only small manually annotated corpus because we want to have a clear point of comparison with previous work. We use the same features as in [10], i.e., bag of words, to simplify discussion. Contrary to [10], there is no need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms, cf [16], [17], [20].

This features are extracted from text in the following process. First a text window surrounding ambiguous word of ± 20 segments (tokens) is constructed.[8] Then the occurrence of a word[9] is noted in a feature vector. Every dimension corresponds to different word. The resulting vectors are sparse. Instead of using raw frequencies we tested a few weighting schemes coupled with *cosine* function for measuring similarities between contexts. The following measures were tested:

- Term frequency, inversed document frequency (henceforth, *tf.idf*), see [20]. We assume, that document is the same as context in this measure.
- *Logent* — values were scaled with logarithm and divided by entropy of a context (standard Shanon entropy counted as $\sum p \log p$, where p is estimated using MLE). This technique was proposed in *Latent Semantic Analysis* by Landauer and Dumais [26].
- Mutual information (definition following work of Lin [27]). We use *lin_cos* for denoting this measure.
- Discounted pointwise mutual information (*pmi*), see [28]. The most important difference between pmi and lin_cos is that pmi uses discounting factor to address the problem of overestimation of mutual information in case of infrequent events.
- Rank Weight Function (RWF) based on mutual information defined by Lin (*rwf_lin*), see [18].
- Rank Weight Function (RWF) based on pointwise mutual information (*rwf_pmi*), see [18].

[8]A segment (token) is defined as word, words separators, but some words can be split onto several segments. For discussion see [25].

[9]A word is a fuzzy concept, more specifically we use a base forms of a word coupled with its flexemic class.

Both RWF function works by using ranks instead of exact feature values. It allows for certain level of generalization from word occurrence frequencies, which can be accidental. RWF approach was previously used in a task of finding similar words in large corpus with very good results [18].

D. Results

Tab. II-VII present results averaged for all the words using different weighting schemes. The first thing to notice is that the results are very hard to grasp. We have noticed several regularities. Firstly, the results of GHSOM evaluation are high in terms of Purity, NMI and Rand Index and very low in terms of F-Measures. Secondly, the amount of data to analyse is very high: 5 weighting measures $\times$ 5 evaluation measures $\times$ 25 algorithm variants. And last but not least, there is no "best" algorithm, i.e., an algorithm that would rank highest in all the different evaluation measures.

To tackle the first problem we will analyse GHSOM in isolation from the other algorithms. We can observe, that the Purity measure for GHSOM has always got the highest values (over 70%). In case of growing hierarchical SOM, we cannot simply define desirable number of clusters, which is strictly determined by number of neurons in each layer. In our — relatively small — set of input data, reading results from all network layers leads to a number of singleton clusters. As was written above, Purity of such clusters equals 1. This fact artificially increases results and clearly shows the weakest side of the Purity measure. Evaluation using F-Measures shows totally opposite results, discarding GHSOM as a efficient clustering method.

There are also some *not available* results (*n/a*) for logent weighting in GHSOM rows. In this particular case, we observed uncontrolled growing of first layer of neural network. Tunning the parameters was also not helpful. The size of the layer reaching 20x20 (which represents 400 clusters), while having data set with only several groups was obviously too big, so we decided to discard such results. The growth of the network might be caused by the nature of logent weighting on this particular dataset. By taking logarithms of feature values the vectors become extremely sparse, because of large number of ones in the features.

One of the most important properties of GHSOM, i.e., not having to specify number of clusters in advance, becomes its most important drawback using standard evaluation measures. To overcome problems with GHSOM two steps can be taken: analysing the resulting network for searching the desired number of clusters or using evaluation measures that can capture other positive properties of GHSOM, most importantly the spatial relations between neurons.

For resolving the second problem with large amount of data to analyse we tested whether some of the evaluation measures are correlated to each other. We used Spearman rank correlation coefficients[10]. The results of this evaluation were

[10]We tested also Pearson's correlation coefficient. The results were also suggesting high correlation among measures, but rank correlation coefficient is more robust to outliers, in our case — GHSOM.

interesting: in most cases correlation was very high between Purity, NMI and RI. Average Spearman's $\rho = 0.83$ and the two-tailed p-value were always lower then 0.0001. Correlation between F_1 and $F(P)_1$ is lower, but also high ($\rho = 0.64$), where usually F_1 is a little bit lower. As $F(P)_1$ allows only for assigning given sense label to cluster only once, we will focus only on decision-based F_1, which scores every pair of clustered contexts.

Reducing the measures to only two (i.e., Purity and F_1) does not solve the problem of choosing the best combination of clustering and weighting scheme. Thus we created two rankings of all the pairs of <clustering algorithm, weighting scheme> ordered by Purity (R_1) and F_1 (R_2). To rank different weighting schemes we use a sum of average of both ranks. This approach can be interpreted as choosing a weighting scheme for which on average the clustering solutions are better then the other. The best weighting scheme is pmi, followed by lin_cos, tf.idf, rwf_lin, rwf_pmi and logent. The difference between pmi using discounting factor and without discounting factor (lin_cos) is minor. The same applies to RWF versions of mutual information based measures. The best results were achieved by agglomerative clustering with weighted average link criterion function while using pmi weighting (Agglo(wslink) in Tab. V). Following are measures using both agglomerative and divisive clustering using (weighted and unweighted) average link criterion for Agglo and e1, h2 and g1 for Rbr. The worst performing algorithms are K-means and K-medoids, which is not surprising—we are using them as another way of setting a baseline.

Those results are interesting, because in related tasks graph partitioning was performing better then the other algorithms. Also, best result achieved for pmi-based weighting scheme is interesting, but not as surprising as these family of measures achieves very good results in the task of finding similar words [18].

The results can be compared to precision of supervised algorithms, because precision corresponds to Purity in our evaluation. All the algorithms beat the baseline of selection always most frequent sense. The best supervised classification algorithm tested on the same dataset, using the same feature set achieved precision of 72.42% as reported by [10], the best clustering algorithm have Purity=71.36%. These results is very high, as usually unsupervised approaches have troubles with beating MFS baseline [2]. This can be explained with high quality of corpus annotations, small corpus size and partially balanced sense distribution in corpus.

TABLE II
AVERAGE RESULTS FOR TF.IDF WEIGHT

	Purity	**NMI**	**RI**	**F_1**	**$F(P)_1$**
Agglo(clink)	49,80	0,137	0,515	41,03	42,43
Agglo(e1)	63,85	0,281	0,679	41,50	50,46
Agglo(g1)	63,03	0,274	0,672	41,66	50,09
Agglo(g1p)	56,90	0,235	0,550	54,49	54,07
Agglo(h1)	57,66	0,243	0,638	41,49	44,57
Agglo(h2)	61,83	0,261	0,662	40,02	48,90
Agglo(i1)	53,36	0,211	0,512	45,37	45,77
Agglo(i2)	63,77	0,292	0,678	42,55	50,69
Agglo(slink)	48,17	0,090	0,385	48,68	45,64
Agglo(upgma)	61,50	0,301	0,637	**58,32**	**58,07**
Agglo(wclink)	49,80	0,137	0,515	41,03	42,43
Agglo(wslink)	48,17	0,090	0,385	48,68	45,64
Agglo(wupgma)	64,19	0,314	0,674	50,13	57,19
Graph	64,65	0,303	0,676	44,51	51,85
K-means	55,28	0,169	0,630	36,33	44,27
K-medoids	52,98	0,141	0,611	39,45	44,05
Rbr(e1)	68,38	0,342	0,707	44,97	53,73
Rbr(g1)	69,88	**0,378**	**0,727**	47,40	55,15
Rbr(g1p)	64,93	0,326	0,662	52,58	57,49
Rbr(h1)	66,00	0,332	0,685	43,88	51,80
Rbr(h2)	68,52	0,348	0,709	45,68	54,10
Rbr(i1)	58,72	0,275	0,561	47,18	50,76
Rbr(i2)	67,10	0,328	0,691	44,85	52,46
GHSOM	**74,55**	0,336	0,668	13,32	29,26
GHSOM – first	62,20	0,276	0,660	42,29	49,18

TABLE III
AVERAGE RESULTS FOR LOGENT WEIGHT

	Purity	**NMI**	**RI**	**F_1**	**$F(P)_1$**
Agglo(clink)	52,36	0,143	0,781	41,19	36,65
Agglo(e1)	51,99	0,142	0,768	44,59	37,98
Agglo(g1)	53,08	0,157	0,776	45,18	40,33
Agglo(g1p)	58,65	0,210	0,662	37,78	45,47
Agglo(h1)	53,38	0,168	0,800	34,96	33,53
Agglo(h2)	53,46	0,163	0,807	32,14	33,95
Agglo(i1)	52,42	0,154	0,777	44,59	39,15
Agglo(i2)	51,65	0,147	0,769	44,59	38,21
Agglo(slink)	52,23	0,145	0,778	43,82	37,80
Agglo(upgma)	51,84	0,143	0,773	43,05	37,69
Agglo(wclink)	51,14	0,126	0,778	40,99	35,53
Agglo(wslink)	52,76	0,151	0,779	43,86	38,11
Agglo(wupgma)	51,16	0,132	0,767	44,32	37,21
Graph	**65,40**	**0,283**	**0,907**	37,81	26,80
K-means	48,02	0,084	0,442	46,18	44,22
K-medoids	48,38	0,099	0,425	**47,65**	44,74
Rbr(e1)	53,70	0,170	0,792	34,53	34,79
Rbr(g1)	55,06	0,190	0,787	44,37	38,92
Rbr(g1p)	60,07	0,222	0,665	37,78	**46,00**
Rbr(h1)	52,40	0,164	0,788	34,43	33,78
Rbr(h2)	52,96	0,170	0,792	34,60	34,83
Rbr(i1)	51,64	0,156	0,782	35,94	34,01
Rbr(i2)	52,35	0,166	0,789	34,36	33,89
GHSOM	n/a	n/a	n/a	n/a	n/a
GHSOM – first	n/a	n/a	n/a	n/a	n/a

IV. CONCLUSIONS AND FURTHER WORKS

This paper presented evaluation of selected clustering algorithms in task of Word Sense Disambiguation for Polish. We have used simple lexical features to represent contexts of ambiguous words. The features were weighted using different weighting schemes. Using pointwise mutual information as a weighting scheme gave best results on average.

Evaluation of clustering was performed on manually disambiguated corpus using five standard evaluation measures [20], [21]. Interestingly, in our settings some of the measures are highly correlated, thus it is only necessary to use two of them, e.g., Purity and F-Measure. This might be caused by the fact, that we know how many clusters are in the data in advance, but one of the problem that the more elaborated measures are trying to address is the problem with generation of large number of clusters.

TABLE IV
AVERAGE RESULTS FOR LIN_COS WEIGHT

	Purity	NMI	RI	F_1	$F(P)_1$
Agglo(clink)	51,29	0,141	0,561	40,58	44,59
Agglo(e1)	63,62	0,300	0,688	42,64	51,87
Agglo(g1)	56,69	0,232	0,563	54,49	54,31
Agglo(g1p)	60,07	0,217	0,670	37,94	47,43
Agglo(h1)	62,95	0,294	0,672	41,99	49,27
Agglo(h2)	62,50	0,261	0,667	40,41	49,41
Agglo(i1)	53,07	0,170	0,476	46,62	46,15
Agglo(i2)	64,37	0,295	0,686	43,03	50,46
Agglo(slink)	48,24	0,099	0,395	49,23	45,94
Agglo(upgma)	61,28	0,291	0,637	**56,79**	56,22
Agglo(wclink)	51,29	0,141	0,561	40,58	44,59
Agglo(wslink)	48,24	0,099	0,395	49,23	45,94
Agglo(wupgma)	67,19	0,344	0,682	51,97	58,48
Graph	65,17	0,306	0,682	45,64	52,83
K-means	54,54	0,162	0,629	35,68	43,30
K-medoids	52,24	0,155	0,608	39,23	44,80
Rbr(e1)	69,57	0,367	**0,722**	46,81	55,44
Rbr(g1)	66,36	0,339	0,657	53,54	**58,92**
Rbr(g1p)	64,44	0,291	0,709	41,91	51,63
Rbr(h1)	67,71	0,337	0,694	44,91	52,54
Rbr(h2)	68,90	**0,368**	**0,722**	46,61	54,77
Rbr(i1)	59,91	0,287	0,576	47,53	52,26
Rbr(i2)	68,15	0,332	0,695	45,07	52,83
GHSOM	**73,51**	0,323	0,666	12,70	28,09
GHSOM – first	56,47	0,210	0,641	39,38	45,77

TABLE V
AVERAGE RESULTS FOR PMI WEIGHT

	Purity	NMI	RI	F_1	$F(P)_1$
Agglo(clink)	50,47	0,133	0,572	39,24	41,68
Agglo(e1)	61,53	0,268	0,661	40,53	48,29
Agglo(g1)	60,03	0,253	0,658	39,85	47,91
Agglo(g1p)	62,58	0,258	0,663	39,83	48,07
Agglo(h1)	65,33	0,303	0,686	43,94	53,30
Agglo(h2)	63,56	0,300	0,686	42,22	50,99
Agglo(i1)	56,85	0,240	0,560	49,08	49,71
Agglo(i2)	64,14	0,280	0,676	42,50	50,55
Agglo(slink)	48,32	0,096	0,391	48,81	45,79
Agglo(upgma)	61,66	0,311	0,641	**56,75**	58,39
Agglo(wclink)	50,47	0,133	0,572	39,24	41,68
Agglo(wslink)	48,32	0,096	0,391	48,81	45,79
Agglo(wupgma)	68,38	0,368	0,710	52,78	**59,46**
Graph	62,27	0,268	0,662	42,50	49,85
K-means	56,32	0,181	0,635	36,83	45,38
K-medoids	53,42	0,155	0,614	40,10	44,87
Rbr(e1)	70,92	**0,399**	**0,736**	48,64	55,60
Rbr(g1)	69,05	0,358	0,721	46,29	53,37
Rbr(g1p)	68,98	0,370	0,723	46,58	54,26
Rbr(h1)	67,19	0,316	0,683	44,03	51,05
Rbr(h2)	71,36	0,393	0,732	48,28	55,82
Rbr(i1)	64,45	0,338	0,646	50,49	56,04
Rbr(i2)	68,07	0,331	0,696	45,06	52,17
GHSOM	**74,71**	0,329	0,667	12,98	28,93
GHSOM – first	59,82	0,241	0,651	41,24	49,02

TABLE VI
AVERAGE RESULTS FOR RWF_LIN WEIGHT

	Purity	NMI	RI	F_1	$F(P)_1$
Agglo(clink)	49,29	0,121	0,578	38,16	42,45
Agglo(e1)	60,57	0,240	0,680	39,35	48,69
Agglo(g1)	52,10	0,133	0,775	43,84	38,19
Agglo(g1p)	55,39	0,204	0,625	38,15	44,40
Agglo(h1)	61,15	0,250	0,680	40,36	48,78
Agglo(h2)	60,07	0,214	0,669	37,89	47,65
Agglo(i1)	57,13	0,215	0,621	42,92	45,90
Agglo(i2)	61,44	0,246	0,676	39,23	47,95
Agglo(slink)	49,35	0,123	0,473	**46,79**	45,07
Agglo(upgma)	58,80	0,221	0,655	42,11	48,17
Agglo(wclink)	49,51	0,121	0,572	38,85	42,52
Agglo(wslink)	49,35	0,123	0,473	**46,79**	45,07
Agglo(wupgma)	60,36	0,226	0,670	38,89	48,34
Graph	61,48	0,248	0,677	42,00	50,24
K-means	57,21	0,185	0,637	37,03	45,01
K-medoids	53,42	0,176	0,563	43,53	43,53
Rbr(e1)	64,12	0,271	0,694	41,13	50,34
Rbr(g1)	54,06	0,182	**0,795**	35,39	34,75
Rbr(g1p)	63,43	0,289	0,687	43,83	52,51
Rbr(h1)	62,55	0,266	0,687	42,15	50,72
Rbr(h2)	63,58	0,280	0,700	41,45	49,96
Rbr(i1)	64,26	**0,301**	0,700	44,62	**53,18**
Rbr(i2)	63,28	0,285	0,701	42,11	50,71
GHSOM	**71,66**	0,298	0,663	11,09	26,65
GHSOM – first	54,17	0,149	0,615	35,02	42,88

TABLE VII
AVERAGE RESULTS FOR RWF_PMI WEIGHT

	Purity	NMI	RI	F_1	$F(P)_1$
Agglo(clink)	51,30	0,137	0,586	39,14	42,80
Agglo(e1)	58,11	0,201	0,658	37,74	46,67
Agglo(g1)	57,36	0,235	0,560	**51,54**	52,08
Agglo(g1p)	58,91	0,219	0,649	41,56	48,45
Agglo(h1)	59,14	0,218	0,661	38,12	46,36
Agglo(h2)	57,29	0,203	0,652	37,14	43,46
Agglo(i1)	58,05	0,236	0,648	42,64	45,25
Agglo(i2)	60,72	0,238	0,670	40,77	47,93
Agglo(slink)	47,90	0,085	0,416	47,94	44,38
Agglo(upgma)	55,81	0,205	0,627	45,52	48,28
Agglo(wclink)	51,14	0,143	0,594	38,44	42,88
Agglo(wslink)	47,82	0,089	0,415	47,91	44,16
Agglo(wupgma)	60,12	0,221	0,664	38,58	46,95
Graph	62,11	0,251	0,674	40,66	49,22
K-means	58,27	0,207	0,649	38,88	46,44
K-medoids	52,60	0,160	0,565	44,26	44,88
Rbr(e1)	60,23	0,219	0,666	37,85	46,53
Rbr(g1)	66,13	**0,333**	0,672	51,53	**56,24**
Rbr(g1p)	60,85	0,227	0,657	39,06	46,08
Rbr(h1)	60,30	0,224	0,666	38,25	45,55
Rbr(h2)	60,30	0,226	0,668	38,46	46,59
Rbr(i1)	61,60	0,241	**0,675**	41,56	49,14
Rbr(i2)	60,32	0,220	0,665	38,30	46,23
GHSOM	**72,24**	0,308	0,664	10,93	27,85
GHSOM – first	54,61	0,167	0,622	36,52	43,59

The results of evaluation are interesting. Divisive clustering algorithm are not worst then agglomerative. Previously used graph partitioning approach is sub-optimal. Not surprisingly, K-means and K-medoids produce clustering of lowest quality.

In the future we are planing to extend this work in a few ways. We want to evaluate some clustering algorithms that were tailored to a problem of Word Sense Disambiguation (e.g., Clustering by Committee [29]). Testing using more elaborated features can give more insight into nature of different clustering schemes. Evaluation of the algorithms on large corpus is needed (f.e., using whole IPI PAN Corpus [22]), because used corpus is small and potentially does not capture peculiarities of large amount of unrestricted texts. Of course, such evaluation has to be performed manually, based on examination of only statistical significant samples of results.

Application-based evaluation of clustering in performing WSD in weakly-supervised settings is also needed. We plan to use LexCSD algorithm for this purpose [9], [10]. This is especially important, as we have used graph partitioning clustering, which occurred to be a sub-optimal according to evaluation presented in this work.

ACKNOWLEDGMENT

Work financed by Innovative Economy Programme project POIG.01.01.02-14-013/09.

REFERENCES

[1] C. Fellbaum *et al.*, *WordNet: An electronic lexical database*. MIT press Cambridge, MA, 1998.
[2] E. Agirre and P. Edmonds, Eds., *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.
[3] A. Kilgarriff, "Word senses," in *Word Sense Disambiguation: Algorithms and Applications*, E. Agirre and P. Edmonds, Eds. Springer, 2006.
[4] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 1–69, 2009.
[5] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computational Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.
[6] R. Mihalcea, "The Role of Non-Ambiguous Words in Natural Language Disambiguation," in *Proceedings of the Fourth RANLP*, 2003.
[7] Z. S. Harris, *Mathematical Structures of Language*. New York: Interscience Publishers, 1968.
[8] T. Pedersen, "Computational approaches to measuring the similarity of short contexts: A review of applications and methods," *South Asian Lang. Review*, to appear.
[9] B. Broda and M. Piasecki, "Semi-supervised word sense disambiguation based on weakly controlled sense induction," in *4rd Int. Symp. Adv. in AI and Applications*, 2009.
[10] M. M. B. Broda and M. Piasecki, "Evaluating lexcsd — a weakly-supervised method on improved semantically annotated corpus in a large scale experiment," in *Proceedings of Intelligent Information Systems*, S. T. W. M. A. Kłopotek, A. Przepiórkowski and K. Trojanowski, Eds., 2010.
[11] G. Karypis, "CLUTO a clustering toolkit," Univ. of Minnesota, Tech. Report, 2002.
[12] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 2001.
[13] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the International Conference on Very Large Data Bases*. Citeseer, 1994, pp. 144–144.
[14] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing maps: exploratory analysis of high-dimensional data," 2002.
[15] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks,*, vol. 11, pp. 574–585, 2000.
[16] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, 2005.
[17] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Machine Learning*, vol. 55, no. 3, pp. 311–331, 2004.
[18] M. Piasecki, S. Szpakowicz, and B. Broda, *A wordnet from the ground up*. Oficyna wydawnicza Politechniki Wroclawskiej, 2009.
[19] R. Forster, "Document clustering in large german corpora using natural language processing," Ph.D. dissertation, University of Zurich, 2006.
[20] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
[21] A. Kulkarni and T. Pedersen, "Senseclusters: unsupervised clustering and labeling of similar contexts," in *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, Morristown, NJ, USA, 2005, pp. 105–108.
[22] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science PAS, 2004.
[23] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computational Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.
[24] R. Młodzki and A. Przepiórkowski, "The wsd development environment," in *Proc. 4rd Language and Technology Conference, Poznań, Poland*, Z. Vetulani, Ed., 2009.
[25] A. Przepiórkowski, "The potential of the IPI PAN Corpus," *Poznań Studies in Contemporary Linguistics*, vol. 41, pp. 31–48, 2006.
[26] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition," *Psychological Review*, vol. 104, no. 2, 1997.
[27] D. Lin, "Automatic retrieval and clustering of similar words," in *Proceedings of the Joint Conference of the International Committee on Computational Linguistics*. ACL, 1998, pp. 768–774.
[28] P. Pantel and D. Lin, "Discovering word senses from text," in *Proc. ACM Conference on Knowledge Discovery and Data Mining (KDD-02)*, Edmonton, Canada, 2002, pp. 613–619.
[29] P. Pantel, "Clustering by committee," Ph.D. dissertation, Edmonton, Alta., Canada, Canada, 2003, adviser-Dekang Lin.