Clustering the results from brainstorm sessions

Master thesis Business Administration

UNIVERSITEIT TWENTE.

Author	Jeroen Hek
Student number	s1143336
First supervisor	Dr. Chintan Amrit
Second supervisor	Dr. Mena Badieh Habib Morgan
Specialization	Information management
Date	30 July 2014

Abstract

This research addresses the problem of clustering the results of brainstorm sessions. Going through all the ideas from the brainstorm session and consolidating them through clustering can be a time consuming task. In this research we design a computer-aided approach that can help with clustering of these results. We have limited ourselves to looking at single words and we identify the different factors that can influence the clustering results. These factors are: (1) word similarity algorithm, (2) dimensionality, (3) cluster count, (4) clustering algorithm, and (5) the evaluation approach. In total we tested six word similarity algorithms, two clustering techniques and three evaluation methods, in order to see which configuration works best for the task. We found evidence that the clustering of these results is feasible, but the results are influenced by the subjective behaviour of human interpreters.

Table of contents

1. Introduction

This thesis addresses the issue of clustering the results of brainstorming sessions in such a way that large quantities of ideas can be reduced to a couple of groups. Generating ideas or solutions is the main purpose of a brainstorm session, however excessively generating ideas can also be a problem for the technique. While brainstorming, there will be always some overlap of ideas or solutions within a group, recognizing this overlap and clustering ideas that are similar can be a labour intensive process, especially when the group size increases. By providing an application that is able to interpret the results of these sessions and cluster them according to their semantic relationship, this research will not only be valuable for organizations, but also researchers the application of word sense disambiguation algorithms on word clusters. It also creates an opportunity for other researchers to experiment with large scale brainstorm sessions.

1.1. Background

In recent years, organizational brainstorm sessions have been unprecedented popular. The idea is to yield ideas or solutions from the collective mind of multiple people. Osborn (1953) defines a brainstorm session as: "*To practice a conference technique by which a group attempts to find a solution for a specific problem by amassing all the ideas spontaneously contributed by its members*" (p. 151). During a brainstorm session a group or an individual applies a creative technique in order to find a solution for a specific problem by fabricating a list of ideas. This technique has found its way into organizations, where it is being used in a collective approach, where not only ideas are being harvested, but also combined or extended on existing ideas. This generation of new solutions or ideas has been proven very valuable for organizations.

All brainstorm sessions start with the question about the composition of the panel. Osborn (1953) proposed the optimum size of this composition is about a dozen, but advocates for an odd number[1]. This is to ensure there is the availability of a majority, thus avoiding the danger of creating two equal sized groups that can obstruct decision-making conferences. Despite the fact that a dozen group members does not sound that larger, the results of these size groups can still be overwhelming. For example, the 6-3-5 brainstorming technique where six members write down three ideas every five minutes. After six rounds the techniques could generate 108 ideas. Osborn (1953) reported that a session with the American Association of Industrial Editors generated over 400 ideas. This was done by four panels averaging about 50 members each. Also smaller panels were reported where 28 members produced over 200 ideas.

[1] Although Osborn (1953) advocates for an odd number group size is twelve members still an interesting size. With a group of twelve it is possible to create the following subgroups: 2x6, 3x4, 4x3, 6x2

The rules of the brainstorming techniques also enhance the creation of new ideas. Below are the rules according to Osborn (1953) which are briefly discussed.

- Suppress criticism – early judgement against ideas must be withheld;
- Quantity is precondition – the increase of ideas means that the probability of fruitful ideas increases;
- Combination and improvement – combining ideas can yield something greater than the sum of the total individual ideas;
- Open minded to unusual ideas – unusual ideas can create new perspectives.

Based on these rules it can be concluded clearly that the technique is all about creating ideas, and ideally as many as possible. Girotra, Terwiesch and Ulrich (2010) found out that hybrid structures work better, compared to a group of individuals. In the hybrid structure, individuals first work individually and then together. The result of this method is that a hybrid structure generated more ideas and is better for identifying the best ideas.

The main problem with brainstorm sessions is the time consuming task of clustering all the ideas or solutions that are generated. These sessions are held with relative small groups, as for increasing the group size will also increase the result count. This increase will further complicate the task of going through all of the ideas or solution in order to identify major clusters. Thus, to improve this process there is a need to automatically cluster ideas which will give the brainstorm sessions supervisors a quick overview about major clusters. These clusters can be used as input for the next brainstorm session.

The output of brainstorms sessions can consist of single words, multiple words or small sentences that describe a particular idea. In this research, we have limited ourselves to looking at ideas which consist of single words. Because of this limitation the choice has been made to look at word clustering. This is a more fundamental research that can be applied to more problems, than only the clustering of brainstorm sessions.

1.2. Research question

The central question for this study is formulated as followed:

"*Is it possible to reliably cluster language independent individual words in a given communication context?*"

In order to answer this central question, the following questions come to mind:

- Which factors influence the clustering process?
- How can the similarity between words be calculated?

Word clustering can be seen as a transformation process. The input will be a list of words and the output a couple of clusters which contain words that are semantically similar to each other. During the transformation several steps will be taken, during these steps various factors can influence the output. Therefore, one of the sub question is to identify which factors influence the transformation process. Another sub question has to do with word sense disambiguation. Words can be ambiguous and have multiple senses which only with context get meaning. In this setting there is no context available, possibly there is a hidden context that the words share with each other. Thus, without presence of context how can we clusters words without knowing the true word sense.

1.3. Scientific relevance

In recent years the field of word sense disambiguation has been rigorously explored, but has barely been applied to word clustering. Some scholars have applied word clustering to corpora in order to cluster words that bear the same specific meaning. The innovative aspect of this research is the clustering of single words that share a particular context without the presence of context. Techniques from the field of word sense disambiguation will be applied to the research question and their individual performance will be measured. This directly leads to the seconds point: how to evaluate the results of word clustering techniques? Ultimately, human subjects are the only ones that can assess the quality of these clusters. Thus, this research will elaborate on the transformation process of word clustering and lay a foundation for this process. In this process several factors will be identified that can influence the results and will test configurations that produces the best results. Human judgment will be incorporated into this design.

1.4. Social relevance

As brainstorming have become more and more recognized within organization as a method to construct or synthesize ideas. Although a good session can yield a high return on investment. Organization will spend a considered portion of time sorting and clustering the results of these sessions. The organizational relevance of this research is to accelerate this process. The artefact of this research will proof how a computer can assist this process. Also, because this research looks at the fundament of word clustering the possibility exists that this can be applied to other problems where at the heart lies the clustering of words.

1.5. Thesis structure

The theoretical part of the research question will be discussed in chapter two. In this chapter the numerous techniques how to compute word similarity and word clustering. The chapter start with a broad discussion of various techniques within the field of natural language processing with respect to the usage of knowledge bases and afterwards will discuss more in-depth the role of word sense disambiguation and clustering. In chapter three the execution of the literature review will be discussed. Followed by the research design used and the implementation of the design to the research problem. The results of this implementation will be presented in chapter four, and in chapter five these results shall be discussed. Finally, in chapter six conclusions will be drawn from both theory and experiments. This chapter will also answer the research question.

2. Theoretical foundation

This chapter describes the foundation for the continuation of this research. It starts out with a broad discussion about machine learning techniques. Followed by a section about word sense disambiguation that highlighted how to calculate word similarity and finally clustering. The process of obtaining all the literature will be discussed first.

At the start of the literature reviews the following scientific search engines were used: Scopus and ScienceDirect. To obtain a fundamental understanding of the topic a broad search for topic related to word clustering was performed. The following search terms were used to find articles: (1) word sense disambiguation, (2) word sense clustering, (3) word clustering, (4) sense clustering, (5) word similarity and (6) clustering. The subject area filter “computer science” was applied to find specific articles. While reading the articles bibliography they were investigated to see which sources they use, articles that were interesting for later usage were collected and added to the literature. The scientific databases were mentioned before being used to see whom have cited the selected articles. These studies were examined and, if found useful, added to the literature.

2.1. Machine learning approaches

There are multiple approaches to address the problem of disambiguation, ranging from methods with a comprehensive body of knowledge or trained data, to methods which do not know the classification of the data in the training sample. Below are the main learning approaches listed.

- Supervised – makes uses of a training set in which each ambiguous word has been manually annotated with a semantic label;
- Unsupervised – does not make use of any training data, which could be because it is not available. The main task of these methods is word discrimination or clustering;
- Semi-supervised – these methods make use of small training set to teach their classifiers;
- Knowledge-based – rely primarily on external sources. For example; dictionaries, thesauri, ontologies, collocation, etc. which are used to infer the word sense of the target word from the context it belongs to.

2.1.1. Supervised

Supervised learning is a machine learning task in which a disambiguated corpus is used to train the algorithm (Manning & Schütze, 1999). The training data typically contains a set of examples. In these examples each occurrence of the ambiguous word is manually annotated with a semantic label. The classifier analyses the training data to produce an inferring function, which can be used to correctly classify new bases based on their context. According to Navigli (2009) supervised learning yields

better results compared to unsupervised methods. However, it costs a substantial amount of manual labour to sense-tag a corpus for training, which can form a bottleneck for the learning method. In the next subsection a couple of supervised methods will be reviewed.

Bayesian classification

A Bayes classifier looks at the words around an ambiguous word in a large context window (Manning & Schütze, 1999). The idea is that each piece of content can contribute potential useful informative to determine the sense of the ambiguous word. The Bayes classifier is a statistical classifier which applies the Bayes decision rule that minimizes the probability of error when choosing a class (Manning & Schütze, 1999). This is done by choosing the sense with the highest conditional probability, thus minimizing the error rate.

Naïve Bayes

The Naïve Bayes classifier is an instance of a Bayes classifier (Manning & Schütze, 1999), which uses a simple probabilistic classifier. The Naïve Bayes classifier is a widely used supervised machine learning method based on the Bayes' theorem. It is widely used due to its efficiency and ability to combine large numbers of features. The method is called Naïve Bayes because of the "naïve" assumption of independence between features. This assumption has two consequences: (1) the first is that structure of the words within the context is ignored, and (2) the presence of one word in the context is independent of another. This assumption that all features contribute independently is clearly not true (Manning and Schütze, 1999). Terms are conditionally dependent on each other. Despite its incorrect assumption the model can be quite efficient. Navigli (2009) describes the inner workings of the model as follows: "*It relies on the calculation of the conditional probability of each sense S_i of a word w given the features f_j in the context*". The word sense which maximizes $\hat{S}$ is chosen as the most as the most appropriate in this context.

$$\hat{S} = \underset{S_i \in Senses_D(w)}{argmax} P(S_i) \prod_{j=1}^{m} P(F_j|S_i)$$

Information theory

The Bayes classifier looks at the context windows around the ambiguous word to determine the correct word sense, while having an unrealistic independence assumption. The information theory algorithm takes a different approach. Manning and Schütze (1999) describe this approach as: "It tries to find a single contextual feature that reliably indicates which sense of the ambiguous word is being used".

Decision list

According to Navigli (2009) a decision list is: "*an ordered set of rules for categorizing test instances for assigning the appropriate sense to a target word*". This list exists of rules which behave like an "if-then-else" statement, where each of the rules is weighted. A training corpus is used to extract a set of features. This results in rules of the kind (feature-value, sense, score) and will be ordered based on their decreasing score. The method is based on '*One sense per collocation property*', which status that word surrounding the ambiguous word provide strong clues about the correct word sense. As like in the Naïve Bayes method the word sense with the highest score will be assigned to the ambiguous word.

$$\hat{S} = argmax_{S_i \in Senses_d(w)} score(S_i)$$

Decision tree

A decision tree is a predictive model used to represent classification rules with a tree structure that recursively partitions the training data set. Each internal node of a decision tree represents a test on a feature value, and each branch represents an outcome of the test. A prediction is made when a terminal node (i.e., a leaf) is reached.

A decision tree, like the decision list, behaves like an "if-then-else" statement. The only difference is the hierarchical positioning of features. Each internal node represents test on a feature, each branch represents outcome of test and each leaf node (terminal node) represents class label. A word sense will be assigned when the leaf node has been reached. Manning and Schütze (1999) stated that the C4.5 algorithm is a popular one for decision trees, although they are outperformed by other supervised approaches.

Support Vector Machine

Support Vector Machines (SVM) constructs a linear hyperplane from the given training set and categorizes the examples to one of two categories, which makes it a binary linear classifier. The SVM model represents a point in space where the gap between the separated categories is as wide as possible. The data points lying closest to the hyperplane are considered to be the support vectors. The model is comprised of two elements: (1) a weight vector, and (2) the bias. If the training set is non-separable slack variables can be used to allow the model to separate the space, thus create a linear hyperplane.

2.1.2. Unsupervised

Supervised approaches make of a training set of examples to train their classifiers. However, there are situations when there is no such information available. For example, specialized domains where

available lexical resources are lacking. The challenge for unsupervised methods is to overcome the lack of resources. According to Manning and Schütze (1999) complete unsupervised disambiguation is impossible. Although, sense discrimination can be performed completely unsupervised. This approach is called '*clustering'*. Manning and Schütze (1999) describe this discrimination task as follows: "*one can cluster the contexts of an ambiguous word into a number of groups and discriminate between these groups without labeling them*". Navigli (2009) adds that these discrimination tasks may not create equivalent cluster compared to traditional sense clusters found in a dictionary. This makes it difficult to evaluate these methods. A solution could be to ask humans to assess the nature of the relationship between members of each cluster. In their purest form unsupervised methods do not make use of any machine-readable resources like dictionaries, thesauri, ontologies, etc.

2.1.3. Semi-supervised

Supervised and unsupervised approaches above are on the extreme side of the line. Where supervised approaches have a training set were ambiguous words are manually annotated with their semantic label and, unsupervised approach does not have any training data. Between these two approaches there are also methods which use a small annotated training set to train the classifier. In this subsection the popular method bootstrapping will be discussed.

Bootstrapping

Bootstrapping is a method to build a classifier with little training data and iteratively improve the classifier's performance. One of the problems to overcome is the lack of annotated and scarcity of data (Navigli, 2009). Yarowsky (1995) describes the method as: "*one begins with a small set of seed examples representative of two senses of a word, one can incrementally augment these seed examples with additional examples of each sense, using a combination of the one-sense per-collocation and one-sense-per-discourse tendencies*".

The annotated data of the classifier grows through including the most confident classification found in the untagged corpus. Resulting in a shrinking training set, until a certain threshold (e.g. iterations) is reached. Yarowsky (1995) advises several strategies to which could form the initial seed:

1. Use words in dictionary definitions – entries from a dictionary already appear in the reliable relationships with the target word;
2. Use a single defining collocate for each class – only use context with have a single definition of a the target word;
3. Label salient corpus collocates – words that co-occur with the target word tend to be indicators of the / a target word sense.

The method avoids the need for costly manually annotated data by exploiting two properties of the human language:

- One sense per collocation – Words nearby the target word strongly and consistently contribute to the sense of the word, based on their relative distance, order, and syntactic;
- One sense per discourse– word sense of a target word is consistent within any given discourse or document;

The advantage of this method is that the addition of untagged data to the label data set (Navigli, 2009). Yarowsky (1995) posit the method is more sensitive, compared to typical statistical sense-disambiguation algorithms, to a wider range of language detail. According to Navigli (2009) one of the disadvantages is the lack for select optimal parameters (e.g. pool size, number of iterations, and number of most confident examples).

2.1.4. Knowledge-based

Knowledge-based or dictionary-based learning exploits knowledge resources when there is no information about the sense of the target word. Resources such as dictionaries, thesauri, ontologies, collocations, etc. are used to infer from context the senses of words (Navigli, 2009). Compared to the supervised approach this approach performance is lower, but has the advantage to cover a wider range due to the usage of knowledge resources. However, recently some researchers reported that knowledge-based techniques match the performance of supervised techniques (Navigli, 2009). In the following subsections several knowledge-based approaches will be discussed.

Selectional Preferences

Selectional preferences exploit the number of meanings that a target word could possess in context. Restrictions are imposed on the semantic classes of words co-occurring with the target words, thus constraining the denotation of the direct object. For example, the word '*ride*' expects an inanimate object as direct object. The direct object must also be denoted as for example '*vehicle*'.

Structural Approaches

With the arrival of lexicons, like WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990), several structural approaches have been developed. These approaches exploit the structure of the concept network within the lexicon and analyze the structure interrelationship between senses based on features. Recently, the structural approach using WordNet has found the attention of many scholars (Voorhees, 1993; Agirre & Rigua, 1996; Altintas, Karsligil & Coskun, 2005; Tsatsaronis, Vazirgiannis & Androutsopoulo, 2007; Sebti & Barfroush, 2008; Kolte & Bhirud, 2009). The reason for this attention

is that WordNet can be used as a machine readable dictionary that beside definitions can also contain a hierarchical structure. In the next chapter the benefits of these functions will be discussed.

Thesaurus-based disambiguation

The method thesaurus-based disambiguation exploits the subject categories supplied by a dictionary or the semantic categorization from a thesaurus. Navigli (2009) describes the method as follow: "*The basic inference in thesaurus-based disambiguation is that the semantic categories of the words in a context determine the semantic category of the context as a whole, and that this category in turn determines which word senses are used*".

2.1.5. Summary

The literature is not really clear on when to use a supervised, semi-supervised or unsupervised approach. If there is knowledge available (i.e., manually labelled test data) then they argue to use the supervised approach. Otherwise you should choose for one of the other approaches. Also, the other parameters come in to play when choosing an approach, namely: time and the kind of problem. As stated by Manning and Schütze (1999), if the kind of problem is the clustering of data, a supervised approach is ideal for solving the problem.

Looking at the performance of the different approaches, it is noticeable that the supervised approach triumphs in machine learning. The performance is almost unmatched, nevertheless this performance comes with a huge cost and which is the obtainment of manually labelled training data. This is a time consuming and expensive task. Recently researchers reported that semi-supervised or knowledge based techniques show promising result for several reasons (Navigli, 2009). First, the more structured knowledge is available, the better the performance of these techniques will be. Second, knowledge-based resources used are increasingly enriched (i.e. the evolution of WordNet). Third, the potential of domain ontologies can be exploited by knowledge-rich techniques.

For this research the used method of machine learning approach, depends on the analysis of the word sense disambiguation techniques. During the selection mentioned earlier, parameters will be taken into consideration. In the following section several word disambiguation techniques will be discussed and compared to another. Depending on the results a method will be chosenfor the techniques and related learning approach which will be used.

2.2. Word sense disambiguation

Disambiguation is an intermediate task, because it is necessary to accomplish certain tasks. First a short description of Word Sense Disambiguation (WSD) will be elaborated on, giving a definition and the application of the task. Next there will be an overview of existing literature regarding WSD will be discussed. Finally, several algorithms will be selected and compared to each other.

Ambiguity is common in the human language, so determining the meaning of a particular word depends on the context the word occurs in. As an example of ambiguity, take the word '*match*', WordNet comes up with multiple meaning to the word:

- Lighter consisting of a thin piece of wood or cardboard tipped with combustible chemical; ignites with friction;
- A formal contest in which two or more persons or teams compete;
- A person who is of equal standing with another in a group;
- Something that resembles or harmonizes with.

For word sense disambiguation it is the task to identify which sense (i.e. meaning) of a word is used in a sentence, when the word is ambiguous. Manning & Schütze (1999) describes the task as: "*The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word*". According to Navigli (2009) the task implies: "*Word sense disambiguation is the ability to computationally determine which sense of a word is activated by its use in a particular context*". Context surrounding the word is used to determine the sense of a particular ambiguous word.

WSD has a broad application. Within machine translation words can have different translations in different sentences. For example a English-Dutch translator could translate the English noun 'full' translated to '*volledig'*(complete) or '*verzadigd'*(saturated). It could also benefit information retrieval system to interpret the given query. For example, the query '*bar*' should the system return information about nearby retail establishments where they serve alcoholic beverages or describe something about the atmospheric pressure. Disambiguation is beneficial and most of the time even necessary for a system when it depends on the meaning of the text being processed.

2.2.1. Approaches

Not only can WSD approaches are categorized as supervised or unsupervised. The approaches belong to one of the two main approaches that have previously been identified: (1) text-based approach and (2) structure-based approach (Altintas, Karsligil & Coskun, 2005, Sebti et al. 2008).

- The text-based approach uses a large corpus or word definitions to collect statistical data in order to calculate an estimated score of semantic similarity;
- The structure-based approach relations and the hierarchy of thesaurus of lexical database, which are generally hard-crafted such as the lexical database WordNet of the Roget's Thesaurus.

In the following subsections several text- and structure-based approaches will be discussed.

2.2.1.1. Edge-based approach

According to Jiang and Conrath (1997) the edge-based approach "*is a more natural and direct way of evaluating semantic similarity in a taxonomy*". The approach calculates the distance between words (nodes) in order to measure the similarity between words. Hierarchical taxonomy, like WordNet, can be used to calculate similarity distance. Obviously, the smaller the distance between two words the more similar they are to each other. Methods penalize hierarchical depth as described below. The higher the shared concept of two words is positioned in the hierarchical tree the less the words will be similar.

Voorhees (1993) used the 'IS-A' function of WordNet to determine the true sense of words in information retrieval systems. The 'IS-A' function returns the hyponym of a sense (i.e. car IS-A vehicle). With the use of only the 'IS-A' function, it could be used to determine word sense, although the degradation of performance is due to the small context window in the query (Voorhees, 1993). It is complicated to determine the correct word sense when there is little or no context to make use of. The usage of WordNet increased due to its semantic relations with the database. The latest versions of WordNet offers relations like: hypernymy/hyponymy, meronymy/holonymy, synonymy/antonymy, entailment/causality, troponymy, domain/ domain terms, derivationally related forms, coordinate terms, attributes, and stem adjectives. Many scholars saw the value of these relationships. As like Voorhees (1993), Wu and Palmer (1994) used the relationship structure from WordNet to calculate sementic similarity. Semantic similarity of two words is calculated based on their path length between the two words. Wu and Palmer (1994) used the follow formula to calculate the similarity:

$$ConSim(C_1, C_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3}$$

- C_3 is the least common hyponym of C_1 and C_2.
- N_1 is the length of the path (number of nodes) from C_1 to C_3.
- N_2 is the length of the path C_2 to C_3.
- N_1 is the path length from C_1 to root.

Wu and Palmer (1994) reported accuracy between 57.8% and 99.45%. The approach applied by Wu and Palmer (1994) is called edge-based. The main premise of this approach is; the shorter the path between two nodes the more similar they are.

This concept of path length between words was also used by Agirre and Rigua (1996) whom introduces the conceptual distance approach. According to Agirre and Rigua (1996) this approach is comparable with the approach of Resnik (1995). It calculates semantic similarity based concepts in the hierarchical of WordNet. The approach focus on nouns and looks at; path length between nodes, depth of the nodes and node that subsumes the nodes and the density of the hierarchy.

$$CD(c, m) = \frac{\sum_{i=0}^{m-1} nhyp^{i^{0.20}}}{descendants_c}$$

Where c is the concept that is at the top of a sub-hierarchy, and $nhyp$ is the mean number of hyponyms per node. Agirre and Rigua (1996) reported accuracy between 53.9% and 71.2%.

Another approach that uses the hierarchy of WordNet is that of Altintas, Karsligil & Coskun (2005). They propose an approach that is comparable to the approach of Wu and Palmer (1994). According to Altintas, Karsligil & Coskun (2005) "*concreteness and abstractness are attributes of concepts which can help us improve our estimations when calculating similarity of concepts*". Path length between words alone does not provide information about their similarity. By including the depth of the word and the depth of the word that subsumes the two words, they are able to differentiate between words to result in a better similarity score. The similarity function is defined as followed:

$$sim(w_1, w_2) = \frac{1}{1 + LenFactor + SpecFactor}$$

Where the $LenFactor$ is:

$$LenFactor = \frac{ShortestLen(w_1, w_2)}{2 \times TaxonomyDepth}$$

The $ShorestLen$ is the shortest path between the two given words and the $TaxonomyDepth$ is the deepest words in the taxonomy. The other parameter in the similarity function is the $SpecFactor$. This is the difference between the specificities of the two words. This is calculated as followed:

$$SpecFactor = \left|\left(Spec(w_1) - Spec(w_2)\right)\right|$$

Where;

$$Spec(w) = \frac{Depth(w)}{ClusterDepth(w)}$$

The $depth$ is the depth of the word in question (i.e. the depth of the word 'Entity' is zero). The $ClusterDepth$ is depth to the node which both words share. Altintas, Karsligil & Coskun (2005) report an accuracy that is better compared to that of Wu and Palmer (1994).

2.2.1.2. Node-based approach

The node-based approach determines word similarity based upon the information-content approach (Jiang & Conrath, 1997). The term Information Content (IC) was posit by Resnik (1995) and calculates the word probability in a given corpus. The node-based approach takes the least common ancestor and calculates its IC value. The concept of information-content will be discussed in depth and several node-based approach will be discussed below.

Resnik (1995) introduced the concept Information Concent (IC). The IC value is the probability that a concept occurs in a given corpus. This value is calculated by the negative log likelihood formula:

$$IC(c) = -log\big(p(c)\big)$$

The idea behind quantifying information concept is based on the appearance of words in a corpus. As the probability of appearance increases the information the words carriers decreases, making the word a more abstract concept. Therefore, infrequent words convey more information compared to frequent words. Semantic similarity is calculated as follows:

$$sim(c_1, c_2) = \max_{c \in S\,(c_1, c_2)} IC(c)$$

The more information two words have in common, the more two words are similar to each other.

Jiang and Conrath (1997) combined the information content approach of Resnik (1995) with the edge-based approach. Next to the log probability of the concepts, the method also uses the least common ancestor (LCA) to calculate the similarity between two concepts.

$$sim(w_1, w_2) = \frac{1}{-log\big(p(w_1)\big) - log\big(p(w_2)\big) + 2 \times \log\big(p(LCA)\big) + 1}$$

Like the above method , Lin (1998) combined the information content approach of Resnik (1995) with the edge-based approach as well.

$$sim(w_1, w_2) = \frac{2 \times log\big(p(LCA)\big)}{\log p(w_1) + \log p(w_2)}$$

Both methods state that if two concepts are identical the outcome will be one, when it is zero the two concepts have no common features. Jiang and Conrath (1997) and Lin (1998) compare their work to that of Miller and Charles (1991), which resulted in a .828 and .834 correlation respectively.

Based upon the Information Content of Resnik (1995) Sebti and Barfroush (2008) propose an approach of calculating semantic similarity using the hierarchical structure of WordNet. According to Sebti and Barfroush (2008) they improved the approach of IC with adding the edge counting-based tuning function and reported an accuracy that is higher compared to that of Wu and Palmer (1994) and Resnik (1995).

2.2.1.3. Corpus-based approach

Corpus-based disambiguation approaches are usually semi-supervised, because they use pre-annotated data that has been made by people to serve as training data. Other corpus-based approaches use large corpora to calculate word distances . These distances, or cosine distances, between words allow the algorithm to say something about word similarity by watching how often they share the same context.

Tsatsaronis, Vazirgiannis and Androutsopoulo (2007) combined the spreading activation network, and applied a weighting scheme to their approach. The weighting was applied to the edges of the network and used the term frequency and inverse document frequency also known as TF-IDF to calculate the weight for the edges. The TF-IDF approach is a statistical approach which is often used in information retrieval and text mining. It looks at the term frequency, which can increase proportionally depending on the document, although the inverse document frequency helps to control for the occurrence of words. Words that occur less are more valuable according to the approach compared to words that occur often. Compared to the approach of Véronis & Ide (1990) the approach of Tsatsaronis, Vazirgiannis and Androutsopoulo (2007) does score a better accuracy, namely between 34.3% and 38.7%. Although the results of the approach look promising the computational complexity is one of the disadvantages of the approach. According to Tsatsaronis, Vazirgiannis and Androutsopoulo (2007) the complexity can be described as: $O\big(n \times k^{l+1}\big)$ which is

higher compared to other techniques. This complexitiy is a result of the network the technique builds, in which it includes almost the entire dictionary. Another neural network approach is suggested by Mikolov, Sutskeven, Chen, Corrado and Dean (2013). They have introduced a continuous Skip-gram model that efficiently learns high-quality distributed vectors that represent a large number of syntactic and semantic relationships. The model predicts words within a certain window surrounding a target word. Distant words are less related to the target word and were given less weight, compared to words that are closer to the target word. Mikolov et al. (2013) found that high dimensional word vectors can be used to answer subtle semantic relationships between words. This will tell if words are similar to each other, but also answer questions like: A is like B as C is to D where D will be given by the system (e.g., France is to Paris as Germany is to X, the system will return 'Berlin' as answer).

Cowie, Guthrie and Guthrie (1992) proposed an approach called Simulated Annealing which determines the word sense of a target word using the complete sentences in which the target word operates. The rational of using the context of the target word is that senses that belong together have more words in common with their definitions compared to senses that do not belong together. The approach is comparable to that of Lesk (1986) only here the context of the surrounding word is used to measure overlap in contrast of the definition of other words. According to Cowie, Guthrie and Guthrie (1992) the results are comparable to that of other co-occurrence approaches.

Several scholar also cluster word using corpora (Purandare & Pedersen, 2004; Deepak, Roa & Deepak, 2006; Sugiyama & Okumura, 2009; Zhu & Lin, 2009; Broda & Mazur, 2010; Martín-Walton & Berlanga-Llavori, 2012). The input of these algorithms is a corpus and the output are word clusters contain words that share a similar context with each other. This approach is highly dependent on the context surrounding the words, otherwise it would not know how words are related or differentiate word senses.

2.2.1.4. Other approaches

Dictionary and thesaurus based

Lesk (1986) proposed an approach in which the definition of a word was used to determine the true word sense of a word. He used the Oxford Advanced Learner's Dictionary of Current English to gather the word sense of several words and calculated the overlap between definitions. He took a ten word window around the target word and calculated the overlap. The word sense with the highest overlap (i.e. words that appear in both definitions) is presumed to be the correct one. For example, 'bank' in context with '*money*' or '*mortgage*' will belong to a financial institute, but if the context is about

'*a slope*' or '*body of water*' the word belongs to steep natural incline. The overlap between the senses and the dictionary definitions is calculated by counting words that occur in both sentences.

$$Score_{Lesk} = (S_1, S_2) = |gloss(S_1) \cap gloss(S_2)|$$

Where $gloss(S_i)$ represents the bag of words in the context surrounding the target word and the dictionary definition. The algorithm assumes the correct sense to be that which has the highest overlap. The accuracy of the algorithm scores between 50% and 70% when applied to a sample of ambiguous words. A disadvantage of the overlap function is its sensitivity to the exact wording of definitions.

The neural network approach by (Véronis & Ide, 1990) combines the use of machine readable dictionaries, spreading and activation models. The latter places words (nodes) into a network that can be described as a neural network. These nodes are connected to each other by 'activatory' links that activate concepts which are related to the words. In addition 'inhibitory' links usually interconnect competing senses of a given word. The system builds a network based upon words which after a while grows to almost the size of the dictionary, because of all the relations word senses contain. The second step is to activate the words which need to be disambiguated. Each node activates their neighboring words until the entire network has been activated. Inhibitory links sends inhibition to one another. This strategy activates related words and deactivates unrelated or weakly related nodes. Véronis & Ide (1990) found that their approach works better compared to that of Lesk (1986). The approach of Lesk fails to disambiguate the right sense with the words: pen and page. According to Véronis & Ide (1990) the approach is more robust, because it does not rely on the occurrence of words in word sense definitions.

Because of the sensitivity of the overlap function researchers looked for other possibilities that could be used to compute semantic similarity. Yarowsky (1992) used the Roget's thesaurus in which he identified and weighted words that could indicate the sense of a word in context. The statistical approach used a window of fifty words surrounding the target word. The correct sense was the one which has the highest sum. Beside the overlap and statistical similarity researchers also looked at the structural similarity between word senses. Yarowsky (1992) uses the Bayes' rule to calculate the probability of the senses and determines the right sense where the sum is the greatest. The formula used is displayed below.

$$\underset{rCat}{ARGMAX} \sum_{w\ in\ context} log\ \frac{\Pr(w|RCat) \times Pr(RCat)}{w}$$

Kolte and Bhirud (2009) continued to build on the work of Lesk (1986). They have stated that the overlap function suffers from the definition size in WordNet, which is too small. In order to improve the approach of Lesk (1986) additional definitions are collected from Hypernymy and Hyponymy. These definitions are added to the bag of words. In addition to the hypernymy and hyponymy functions Kolte and Bhirud (2009) also research the impact of other WordNet function, namely: Meronymy / Holonymy, ability link, capability link and function link, although they do not report their findings of these functions. Their disambiguation approach has an accuracy of 63.92%.

Semi-supervised and supervised approaches

There are also approaches with could be categorized as semi-structured due to their usage of sources. Sources like: NY Times corpus, Brown corpus, Wikipedia or other dictionaries. These approaches train their classifier on these data sources and then present the classifier with a words or sentences to disambiguate. The approach from Yarowsky (1995) is a prime example for a semi-supervised approach. The approach uses a large corpus to train their classifier. From the corpus the classifier learn the context from words by looking at the window around the words. These windows are stored and used to compare against new context windows from which the classifier can determine the true sense of the target word. According to Yarowsky (1995) this semi-supervised approach scored surprisingly well, reporting accuracy scores around 72% and displays scores that are similar to supervised approaches.

Supervised disambiguation techniques make use of manually labelled test data. As stated before, this need for manually labelled test data is one on the disadvantages, as is it hard to obtain these resources, although some scholars reported very high scores using this approach. Like Ng and Lee (1996), they have trained their classifier with a corpus in which the senses were pre-tagged with the correct senses. When given unseen sentences the classifier matches it against the training data and uses the best match to determine the sense of the target word. The difference between pre-tagged and untagged data is directly visible in the results the classifiers produce. The supervised classifier score 89% accuracy, which is much higher compared to that of Yarowsky (1995). However, as mentioned before, this difference in performance comes at a price and that price is the time consuming task of pre-tagging training data. Some scholars also tried to combine supervised and unsupervised techniques. Agirre, Rigau, Padró and Atserias (2000) used lexical knowledge sources and combined these with supervised training corpora to enhance their technique. The unsupervised technique alone had an accuracy between 41.6% and 43.6%, when this was combined with the supervised technique accuracy improved and they reported accuracy between 62.0% and 68.3%.

Another example is the approach of Schütze (1997) where news data from multiple sources was used to train their classifier. Similar to Yarowsky (1995) approach it is automatic and can be supplied untagged news data to train the classifier. The approach creates word vectors from words that are close neighbors within the training corpus and neighbors that co-occur within the large window surrounding the target words. Similarity is then calculated by measuring the cosine between these two vectors. This cosine measure can be compared to the normalized correlation coefficient.

Wikipedia

Another knowledge source that has recently caught the attention of researchers is Wikipedia. Wikipedia is a multilingual free internet encyclopedia that is maintained by volunteers. Anyone can access the site and edit almost every article. At the moment Wikipedia contains 30 million articles in 287 languages and the English section of Wikipedia has over 4.4 million articles. An article typically defines and describes an entity or event and has hyperlinks to other entities or events. Mihalcea (2007) uses the hyperlinks within Wikipedia. For example, the word bar could mean law or music and are stored in the articles as piped links (i.e., [[bar (law)|bar]] or [[bar (music)|bar]]). Mihalcea (2007) states that these piped links can be regarded as sense annotations for the corresponding concepts and the content of the articles as the context of the concept. Wikipedia also has disambiguation articles which consist of links to different articles that define different meanings of a concept. Although, these pages were not used due to their incompleteness and inconsistency how disambiguated pages are been used[2]. The first step of the disambiguation process is to collect the piped links of the target word and the related article. In the final step the window around the target word is used to calculate co-occurrence against the Wikipedia articles to determine the correct word sense. Mihalcea (2007) reported accuracy between 79% and 85%.

2.2.2 Summary

As stated by Ide and Véronis (1998), word sense disambiguation is an intermediate task, which means that it is a necessary task to perform in order to accomplish another task. This is also the case with the artifact. First, the correct sense of a word needs to be determined, otherwise clustering of these words would become difficult. It has to be noted that disambiguation is not the primary goal is of this research and other techniques can have a different effect on the cluster results.

The main problem in this research with regard to disambiguation of single words is the lacking context which can be used to determine the sense of a word. Most techniques discussed in this

[2] The inconsistency of disambiguation pages comes from the usage of these pages. For example, the word 'bar' directly goes to the disambiguation page, but the word 'paper' goes to the article that describe paper as: "*thin material produced by pressing together moist fibers, typically cellulose pulp*". The disambiguation article of paper can be found by the identifier 'Paper_(disambiguation)'.

thesis make use of a window surrounding a target word. The reason is that the word sense can be identified by the company the word is surrounded by. This lacking context will definitely have an effect on the accuracy of the disambiguation process. However, the question asked during a brainstorm session will give certain direction to a particular context , this context will be hidden in the relation the words have with each other. Therefore it can be stated that there will be a latent context within the results of a brainstorm session.

Several techniques have been discussed in the section above. Edge-, node- and corpus-bases techniques are the most commen used in the field of word sense disambiguation. All approaches use different resources to train their classifier and achieve different results. Resulting in no clear answer to the answer of which resource provides the best result. Overall, it can be stated that supervised techniques perform better compared to unsupervised techniques. The knowledge-based techniques score relative high and, as stated before, almost match the results of supervised techniques.

The spreading activation network models show real potential despite their complexity. Parameters such as tuning, thresholds and decay make the techniques difficult to use. Another disadvantage is the high processing complexity. Compared to other techniques the network model uses more resource. For these reasons the spreading activation network techniques will not be used in the artifact.

Semantic similarity techniques seem interesting due to their simplicity and reported accuracy. Several WSD algorithms result in almost similar scores, but none of these have ever been applied in the task of word clustering. Therefore, several techniques will be selected and compared to each other in order to measure performance during the clustering task. The edge- and node-based approaches provide very promising results and of these approaches the following will tested:

- Edge based
    - Wu and Palmet (1994)
    - Altintas, Karsligil & Coskun (2005)
- Node based
    - Resnik (1995)
    - Jiang and Conrath (1997)
    - Lin (1998)

From the corpos-based approach the technique of Mikolov et al. (2013) will be used. The authors did not perform any WSD experiments, but it shows promising results for word similarity. Especially with the simplicity of the program. The program reads gigantic files and calculates in high-dimensional

vectors in order to determine what the cosine distances are relative to other words. These vectors can also be used to measure word similarity. The advantage of this approach is that it is not tied to a vocabulary that is pre-determined by other people (i.e. WordNet and other dictionaries). This allows the application domain to read specific texts to expand its vocabulary.

When looking at the evaluation of the different techniques the conclusion can be made that none of them are really applicable to this study. Most of the edge- and node-based techniques use the research from Miller and Charles (1991). They asked 38 students from the University of New York to rate 30 noun pairs on their similarity. They were instructed to give a rating from zero to four, where zero represents no similarity and four perfect similarity or synonymy. This resulted in a list that represents the semantic similarity between the noun pairs. The edge- and node-based algorithms use this research to calculate correlation between their similarity rating and that from Miller and Charles (1991). The word-sense induction techniques compare their results to pre-annotated data, like SENSEVAL[3] (renamed to SEMEVAL) or other knowledge bases. Others used humans to evaluate the findings of their research, because when it comes to language humans are to only subjects to set the golden standard. Because of the lack of an established evaluation metric an fitting evaluation procedure needs to be designed in order to capture and compare performance for each of the algorithms. Advisable is to make use of human informants, because of their skill to determine if words belong to the correct cluster (Gale, Church & Yarowsky, 1992).

2.3. Clustering

In this chapter we will investigate several cluster techniques in order to answer the sub-research question: "*In what way can cluster techniques be applied?*". In the following subsection several techniques are identified and reviewed to see under which conditions they work, also their advantages and disadvantages of each method will be discussed.

For example, if we cluster English nouns according to their syntactic and semantic environment the days of the week will end up in the same cluster, because they share the environment 'day-of-the-week'. Another method to measure word similarity is by looking at the distribution of the neighbouring words, for the two target words and calculates the degree of overlap. Manning and Schütze (1999) advice to spend some time getting familiar with the data at hand, and state it is a mistake to skip this part. When there is no pictorial visualization for linguistic object, clustering could be an important technique in exploratory data analysis (EDA). Clustering techniques can be differentiated into two categories, namely: (1) soft clustering, and (2) hard clustering. Hard clustering techniques allocate each individual object to only one cluster, where soft clustering allows objects to

[3] http://www.senseval.org

be a member of multiple clusters. Mostly clustering techniques use hard clustering, but according to Manning & Schütze (1999) most soft clustering techniques allocate object to only one cluster. Hard clustering techniques have to deal with uncertainty if the assigned cluster is truly the correct one for each object. There always exists the possibility individual object belong to more than one cluster. A true multiple assignment technique is called '*disjunctive clustering*', in object can be assigned to several clusters and also truly belong in those clusters. Manning & Schütze (1999) identified two main structural categories where clustering techniques belong to, namely: (1) hierarchical clustering, and (2) flat clustering techniques. With flat clustering objects are distributed over several clusters, although the relationship between the clusters could not be determined. In hierarchical clustering each node stands for a subclass of their mothers' node, where the leaves of the tree represent elementary clusters. In the next subsections we will discuss several methods belonging to the two main structural categories. Evaluation of clustering results will be discussed in 3.3.1.

2.3.1. Hierarchical Clustering

The bottom-up algorithm, also called agglomerative clustering, start with putting each individual object in a separate cluster. Each iteration clusters two most similar clusters which are determined and combined into a new cluster. This process continues until one large cluster has been created containing all objects. Compared to the bottom-up algorithm, the top-down algorithm, also called divisive clustering, start with placing all the individual object in one large cluster. Each iteration the algorithm determines which objects least similar and splits these objects into new clusters. This continues until each individual object has been separated from their parent cluster. At the end of these algorithms the root of the tree exists of one large cluster which holds all objects, the branches are the clusters, and the leaves exists out of the individual objects.

The algorithms discussed above are part of collection methodologies which describe a particular way of hierarchical clustering. The full list of methods is:

- Single-link – most similar objects are clustered together, like the agglomerative clustering;
- Complete link – least similar object are clustered, like the divisive clustering;
- Group-average – by measuring mean distance between objects and clusters.

In the following subsections these methodologies will be discussed.

Single-link clustering

The single-link cluster follows the same clustering technique as the agglomerative clustering. Objects which are most similar are cluster together until a cluster is created which contains all of the individual object. This technique is closely related to the minimum spanning tree (MST) (Manning &

Schütze, 1999). The MST connects objects with span equal to or less than every other spanning tree. Thus, the smallest span between two objects equals to the largest similarity. The advantage of the single-link clustering is its complexity, which is $O(n^2)$, because every object has to be examined at least once.

Complete link clustering

Where single-link cluster is looking at the maximum similarity between two objects, the complete link cluster uses the negative of this function, thus the minimum similarity. The technique could use the MST, although instead looking for objects which tree spanning is the smallest, complete-link clustering removes objects with the largest span from the cluster. The disadvantage of this technique is its time complexity $O(n^3)$, since there are n merging steps. Each step of the algorithm requires $O(n^2)$ to find the largest distance between two object.

Group-average clustering

A compromise between the single-link and complete-link clusters is called group-average clustering. This technique does not look at data points which are closest and/or furthest separated from each other, but the clusters are based on the average distance between objects within two clusters and merging those which are closest related. The time complexity is the same as that of the single-link technique, thus $O(n^2)$. This could be $O(n^3)$ when the average similarity is calculated each time two group were merged.

2.3.2. Non-Hierarchical Clustering

Non-hierarchical cluster methods create cluster by distributing objects into clusters from a data set which (generally) have a non-overlapping group and no hierarchical relationship between them. An advantage of non-hierarchical methods is the lower demand on computational resources. Compared to hierarchical methods the time-complexity will typically be $O(n)$ or $O(n^2)$. Although, most methods employ multiple passes to reallocate object and refine the result, which will put pressure on the resources. This also raises the question: If multiple passes are needed, when does the algorithm stop working? According to Manning & Schütze (1999) there is one possibility which can be used to determine the stopping criteria is the validity of the cluster's quality. The group-average similarity method can be used to measure this quality. Beside of the validity of quality's problem, there is also the question of the right amount of clusters.

In the following subsections two popular algorithms will be discussed, namely: (1) K-Means, and (2) Expectation Maximization (EM).

K-Means

Manning and Schütze (1999) define the K-means algorithm as: "*a hard clustering algorithm that defines clusters by the center of mass of their members*". The algorithm needs an initial cluster count k which needs to be defined by the user. From the data set k data points are used by the algorithm as initial means. Next an iteration process start where each of the iteration objects are assigned to their closed cluster. After every object has been assigned, a new cluster mean, or centroid, will be calculated. A centroid is the average of all members belonging to a specific cluster, also known as a hypothetical cluster object, although these averages are sensitive for outliers. Manning and Schütze (1999) stated that the use of medoids are less sensitive for outliers, for this in one of the objects within cluster. Also, instead of using the sum of the squared Euclidean distance, K-means algorithms using medoids minimizes the sum of pairwise dissimilarities. The time complexity is $O(n)$ because each iteration distance from each object to the cluster center is calculated and minimal distance determines the object membership of a cluster. A problem arises when objects have equal distance to multiple clusters. One solution is to randomly assign the object to one of the clusters, although this has the disadvantage that the algorithm may not converge (Manning & Schütze, 1999).

Expectation-Maximization (EM)

Compared to the K-means algorithm is the EM algorithm a soft clustering method. The task of the algorithm is to find the maximum-likelihood estimate of the hidden parameters of an underlying distribution (Manning & Schütze, 1999). For each object the probability that it belongs to a cluster and the one with highest probability will be the cluster that object is assigned to, but the object will still have a non-zero membership to all the other clusters. The algorithms follow an iterative process where it calculates the expected membership probability under given parameters and in the second step tries to maximizes this quantity. Advantages of this technique is the possibility of multi-membership. Each object has a certain probability that it is a member of a particular cluster. Compared to a hard clustering method where each object is a member of a single cluster. Disadvantage of the technique is the preparation of parameters that makes it loose its simplicity compared to alternative techniques (Christophe, 1997).

Density Based Spatial Clustering of Applications

The k-means and EM clustering algorithm produce circular shaped clusters based on distance, but are incapable to detect arbitrarily shaped clusters. The density based spatial clustering of applications (DBSCAN) can discover these arbitrarily shaped clusters in a data space (Han & Kamber, 2006). DBSCAN is a density-based clustering technique, that clusters creates clusters from regions that have a sufficiently high density. According to Han & Kamber (2006) a cluster is defined as "*a maximal set of density-connection points*". The techniques searches for clusters by checking the neighbours of

each object. If in the neighbourhood are more that the defined minimal points are present the technique creates a core object. DBSCAN iteratively collects the directly density-reachable object from the core object. The process stops when there are no new objects that can be added to any cluster.

Advantage of this technique is that it can detect arbitrarily shaped clusters. It is non-deterministic meaning and can detect the number of clusters by itself. Through the process of identifying core objects the technique is able to determine by itself how many clusters there are in the given data. Ester, Kriegel, Sander and Xu (1996) demonstrated that the DBSCAN techniques is an attractive approach for the task of identifying clusters in a spatial database. They demonstrated that DBSCAN is significantly more effective in identifying arbitrarily shaped clusters compared to CLARANS. Also, DBSCAN outperforms in terms of efficiency.

2.3.3. Summary

With respect to the complexity, the k-means is at first sight an advantage compared to hierarchical clustering. However, because the method is non-deterministic in the assignment of cluster centroids, has to be performed several times to make sure the assignment of centroids is correctly instead of randomly assigned. The multiple executions of the k-means method will require more computing power than original described, but is still significantly lower than the complexity $O(n^3)$.

The hierarchical clustering has the advantage that it will create clusters from one to N, where N is the number of objects given to the clustering algorithm. Usually the algorithm stops if there is only one cluster left (single-link), or when the number of clusters are equal to that of the number of objects (complete-link). However, it is also possible to configure the algorithm to stop at a specific number of clusters.

The advantage of the EM algorithm is that it is a soft clustering method so objects can be members of multiple clusters. The down-side of the EM algorithm is its sensitivity to the initialization of its parameters. If the parameters are not initialized well the algorithm usually gets stuck at one of the many local maxima that exists in the space (Manning & Schütze, 1999). This initiation could be resolved by first performing the k-means algorithm and using its results to initiate the parameters.

The literature is not clear for which of the clustering techniques is better, so the influence of different techniques on the clustering results will have to be taken in regard. Therefore, different techniques will first be tested to see how much they differ from each other. The goal in this research is to find clusters with a shared context. Therefore, hard-clustering techniques will be used to obtain these context clusters.

3. Research question and approach

This chapter describes how the research is addressed methodically. It serves as a bridge between theory and practice. First, the significance and objective of the research will be discussed. Followed by, explanation of the literature review and the research design and methods that will be applied to address the research question will be described. Finally, the application of the research design to this particular problem will be addressed. In this section will addressed how the research will actually be conducted.

At the moment word sense clustering is only performed with the usage of corpora (Purandare & Pedersen, 2004; Deepak, Roa & Deepak, 2006; Sugiyama & Okumura, 2009; Zhu & Lin, 2009; Broda & Mazur, 2010; Martín-Walton & Berlanga-Llavori, 2012). Their approach is to find words that belong together and cluster them. Finding semantically related words is done by looking in the context they are found in. When two words share the same context it can be assumed that they are related to each other. Most of these researches were done on the English language, but some also applied the algorithms on other languages: Chinese (Zhu & Lui, 2009) and Polish (Broda & Mazur, 2010), but word clustering without the usage of an corpus has not be done before.

The problem that this research addresses is:

"*Is it possible to reliably cluster language independent individual words in a given communication context?*"

The objective of the research is to explore techniques that can cluster words that only share communication context. As stated in the introduction we limit ourselves to single words, thus the input of our method will be a list of single words. We cluster the words according to their semantic relationship. Resulting in clusters that contain words which are semantically related to each other. The challenge in this research is to find related words without the presence of context. It is clear that without context it is difficult to determine the sense of a word and measure how related two words are to each other. Thus, it can be expected that clustering without context will score lower, compared to when context is available. Another challenge is language. The goal is to cluster Dutch words and as seen in the paragraph above other scholars are able to apply the algorithms onto other languages. However, a lot of the research has been performed on the English language, the available knowledge bases are comprehensive. Less comprehensive knowledge bases could influence the performance of the techniques.

3.1. Literature review

In this section will be discussed how the literature that is published related to the research question is gathered. Scopus[4] was the main database used. Before looking at the titles the following filter was applied on the subject areas: "Computer Science". Then the following steps were taken: (1) selecting potential articles by title, (2) from the subset selecting those that seem interesting after reading the abstract, and (3) finally selecting those articles that fit the research question after reading the full text. This filtering process is displayed in the table below. Additional articles were selected by looking at papers that were cited by the scholar, and those that cited the article of the scholar.

Search term	Result	Filter	Title	Abstract	FullText
word sense disambiguation	1380	1052	119	54	31
word sense AND clustering	243	178	39	19	6
word sense induction	85	43	21	8	6

Table 1 Literature search results

Hence, an examination was carried on ways to disambiguate word sense and this resulted in the following result. The table below identifies four main approaches of word sense disambiguation. These approaches will be researched for their applicability.

Article	Edge-based	Node-based	Corpus-based	Dictionary-based
Lesk (1986)				x
Véronis & Ide (1990)				x
Cowie, Guthrie and Guthrie (1992)				x
Yarowsky (1992)				x
Voorhees (1993)	x			
Wu and Palmer (1994)	x			
Resnik (1995)		x		
Yarowsky (1995)			x	
Agirre and Rigua (1996)				
Ng and Lee (1996)			x	
Jiang and Conrath (1997)		x		
Schütze (1997)			x	
Lin (1998)		x		
Agirre, Rigau, Padró and Atserias (2000)			x	
Altintas, Karsligil & Coskun (2005)	x			
Tsatsaronis, Vazirgiannis and Androutsopoulo (2007)			x	
Sebti and Barfroush (2008)		x		
Kolte and Bhirud (2009)				x
Mikolov, Sutskeven, Chen, Corrado and Dean (2013)			x	

Table 2 shows the conceptual matrix derived from the literature review

[4] http://www.scopus.com

3.2. Research design

According to de Vaus (2001) "*the function of a research design is to ensure that the evidence obtained, enables answering the research question as unambiguously as possible*". Design science has been selected as the methodology used for this research. The design science research methodology according to Hevner et al. (2004) "*creates and evaluates IT artifacts intended to solve identified organizational problems*". Such artifacts include implemented or prototype IT systems (instantiations), algorithms or practices (methods), abstractions or representations (models) or constructs. The design science could be viewed as a conceptual process which helps researchers to legitimize research. The design process exists out of two important activities: (1) build, and (2) evaluate. These two activities form a loop within the research design and typically multiple iterations are done before the final artifact produced. During the build activity resources are required to realize the objectives of the research. These resources typically contain knowledge of the theory which contributes to the design. Thorough evaluation is crucial, because of the artifacts purpose to solve a specified organizational problem.

The output of this research will be a method that can be applied to a broad possibility of word clustering tasks. The research methodology helps to design this method in a structured fashion. First, we will address the seven guidelines proposed Hevner et al. (2004) which helps researchers to conduct, evaluate, and present their design science research. Followed by the model Peffers et al. (2008) developed for the production and presentation of IT artifacts in a design science process.

3.2.1. Design science guidelines

According to Hevner et al. (2004) "*the fundamental principle of design science research from which our seven guidelines are derived is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact*". They proposed the guidelines to help researchers, reviewers and editors to understand the methodology of performing effective design science. These guidelines are:

1. **Design as an artifact** – The result of design science research is an artifact, which purpose is to address an organizational problem. Artifacts are rarely full functional information systems. Instead, the artifacts can be viewed as innovations or proof-of-concepts that could represent a crucial effective solution of the identified problem;
2. **Problem relevance** – The main goal of design science research is to develop an artifact, based on a rigor understanding and comprehensive knowledge. This artifact enables the organization to overcome organizational problems or be applied to exploit opportunities;

3. **Design evaluation** –A crucial component of the design process is evaluation. Data needs to be gathered and analyzed with appropriate metrics in order to fully evaluate the artifact. This artifact is completed when the requirements, which it was meant to solve or exploit, of the problem or opportunity respectively, are met.
4. **Research contributions** – Effective research should make contributions in several areas through: (1) the artifact which is meant to solve or exploit the organizational problem or opportunity, respectively. (2) Foundation, scientific contributions through the extending or improvement of existing foundations. (3) Methodologies, the use of development and evaluation methods.
5. **Research rigor** – During the process, rigor must by applied during the construction and evaluation of the artifact. Theoretical knowledge and research methodologies are used to derive rigor. Appropriate theory is used to develop the artifact and to appropriately evaluate the artifact;
6. **Design as a search process** – The process is essentially a search available means to create an effective solution or exploitation of the problem or opportunity. This process is iterative to generate an artifact, evaluates this against the requirements and continues until these requirements are met;
7. **Communication of research** – The result must be presented to two audiences: (1) technology-oriented and (2) management- oriented. These audience have different needs. Technology-oriented audiences need more detailed information which describes the construction and evaluation of the artifact within appropriated organizational context. The management-oriented will determine which organizational resources need to be committed and in which organizational context the artifact is going to be used.

With these guidelines Hevner et al. (2004) is trying to prevent researchers whom apply the design science methodology to fall in the pitfall of overemphasizing on the technological artifact instead of maintaining an adequate theoretical foundation. If there is too much focus on technology it could become a well-designed technical artifact which is not applicable in an organizational setting.

3.2.2. Process model for the research

Peffers et al. (2008) designed a nominal process model which is strongly based on the guidelines of Hevner et al. (2004). The process model incorporates six steps: (1) problem identification and motivation, (2) definition of the objectives of a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication. It is clear that Peffers et al. (2008) recognized the importance of the build and evaluate activities. The process of this model can be viewed in the figure below.

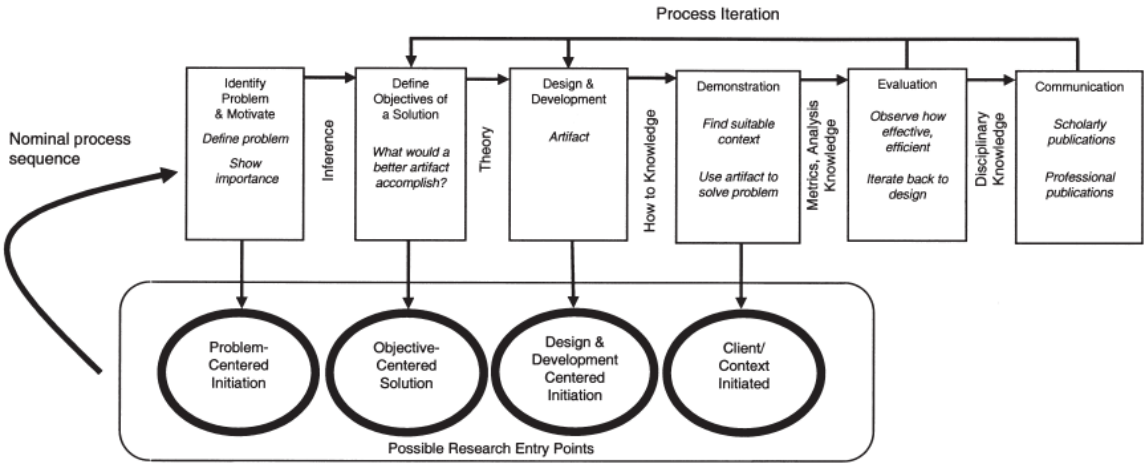


Figure 1 Peffers et al. (2008) Design science process model

The activities of the process model are briefly described as:

1. **Problem identification and motivation** – In the first act of the model the research problem is specified and the value of the solution is justified. The first helps by effectively addressing the problem, where the latter helps the audience of the research to understand the researcher's reasoning.
2. **Define the objectives for a solution** – After the research problem is specified the researchers can infer objectives from this problem definition. These objectives will help to understand what is possible and feasible.
3. **Design and development** – This activity involves the creation of the artifact.
4. **Demonstration** – After the artifact is created it will need to demonstrate it can actually solve the initial problem.
5. **Evaluation** – Observing and measuring how well the artifact solves the problem is needed to consider if the artifact has to be to changed or improved. If the choice has been made to alter the artifact the researcher returns to activity 3 as part of an iterative process.
6. **Communication** – In the final activity of the model the researcher will communicate its findings to the research audience. These findings will have to be communicated to differently oriented audiences, where different needs should be taken into regard.

3.3. Applying the methodology to the research problem

The research focusses on the question of how to cluster the ideas from brainstorming sessions. It is therefore based on a 'problem-centered solution', thus starting at the first activity in the process model. The following sections describe how the design science process model of Peffers et al. (2008) is followed in the course of the research. Because problem identification and objective definition are already discussed at the start of this chapter, applying of the methodology will therefore start at the design and development of the artifact.

3.3.1. Design and Development

From the theoretical framework several process steps have been identified. With these steps the following method have been synthesized:

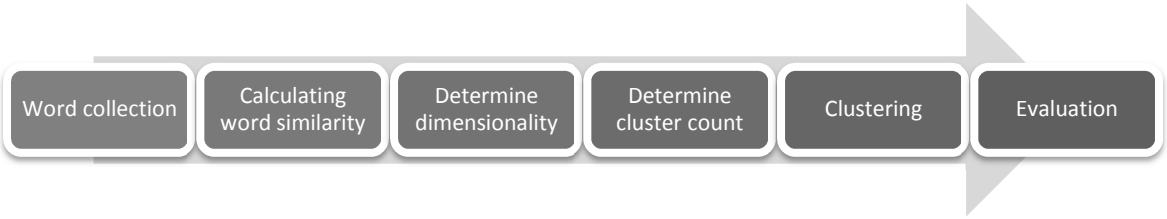


Figure 2 Word clustering process method

The actual implementation of each step can differ. For example, for the process step: "*calculating word similarity*" six algorithms have been identified that are being tested. In the following subsections the implementation of each step will be discussed. During the execution of the method the search will be for the configuration that yields the best results.

Word collection

Normal approach to cluster words is to make use of corpus that serve as input for the algorithm. However, this is not possible in this research, because input for the method is a list of single words. Three different collection approaches will be applied. First, one hundred words will be selected from a dictionary, these one hundred words which form ten word clusters. These words were selected by the IS-A relation they share with a common ancestor in Cornetto (Vossen, Hofmann, de Rijke, Sang & Deschacht, 2007), (i.e. animals, vehicle, music instruments, etc). In figure 3 an example of this IS-A relation is given. The word coast is a subtype of a shore. The higher the word is positioned in this ontology the more abstract and less information the words carries. Thus, the leaves of an ontology carry more information and can be self-explanatory of directly describe the context where they are found in.

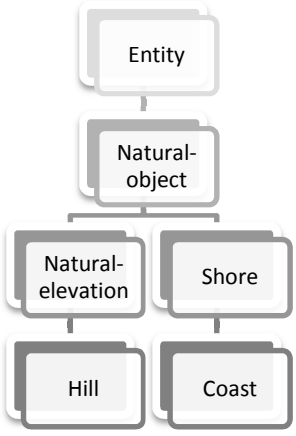


Figure 3 Is-a fragment of the ontology WordNet

Second, usage of human informants, a group of people will be asked to answer the following question: “Welke activiteiten onderneem je in het weekend?” (Which activities do you embark on during the weekend?). After gathering the answers, the clustering process will be applied to the given answers and the results are given back to them. When presented with the clusters they have the opportunity to correct words that have been wrongly clustered. Third, multiple smaller groups will be asked to answer the same question as before. Only now they are asked to clustered the given answer. During the first word collection approach multiple configurations are tested to see which performs best and reduces the amount of configuration that we want to test with human informants. This latter because it is a time consuming task to test each configuration while using informants, thus by reducing this amount we can eliminate less favourable configurations. The second and third approach makes use of humans in two ways: (1) as suppliers of existing knowledge, (2) as judges. According to Gale, Church and Yarowsky (1992) humans are the only ones that can judge word clusters. As the suppliers of existing knowledge it is the task for the algorithm to replicate these results as closely as possible.

Calculating word similarity

During this stage various word sense disambiguation algorithms will be used to calculate word similarity. Each of the algorithms calculates similarity for all possible word combinations. The result of this stage will be a symmetric matrix. In the following subsections the applications of each similarity approach will be discussed and which steps need to be carried out in order to get the algorithm to perform.

Edge-based

For the edge-based approach the Dutch version of WordNet will be used, namely: Cornetto (Vossen et al., 2007). The database combines the Dutch part of the EuroWordNet with the “*Referentie Bestand Nederlands*” and alignes this with the English WordNet to form a formal ontology.

To compute the similarity between words first the true sense of the words have to be determined. This is done by looking at the context the words are in, and based on the other words surrounding the particular word the true sense will be selected where the similarity between words is the highest. This approach will be used for each word to determine its true sense. Similarity between words shall be calculated between the true sense of the words. This approach is used with the following algorithms, also their abbreviation is given what will be used during the experiments to identify each algorithm: (1) Altintas, Karsligil and Coskun (2005) – AKC; (2) Wu and Palmer (1994) – WUP.

Node-based approach

As discussed earlier the node-based approach is based on the edge-based approach, but combines this with information content. In this approach Cornetto will also be used as an ontology. To compute the information content the Twentse News Corpus has been used. The original corpus consist of 10,621 xml files containing Dutch national newspapers, television subtitle and teleprompter (auto-cues) files. Each xml file was stripped of its xml tags and merged forming one document containing all the words of the news corpus. Resulting in one single file containing 321 million words. This file was used to calculate each word probability and stored, so that it could be used by the various approaches. The approaches that apply the node-based approach are: (1) Resnik (1995) – RSK, (2) Jiang and Conrath (1997) – JNG, and (3) Lin (1998) – LIN.

Corpus-based approach

For the corpus-based approach the program of Word2Vec was used to compute vectors representing words. The corpus-based approach also uses the Twentse News Corpus to compute its word vectors. First all vectors were computed based upon the corpus, this resulted in a file containing the cosine distances to neighbouring words. To compute the vectors the default settings of the program were used. The vector file was used by a Python script[5] that contains a function to calculate similarity between two words. The original program was designed to display cosine distance, create words classes (i.e., words like carnivores, coyotes and crocodiles were put together to create an animal cluster), and answer word phrases questions like: if A is to B then is C is to D. Originally, the program did not have the ability to measure similarity, therefore the Python script was chosen. During test of the Python script, it had trouble managing large vector files. With files above 100 million words the program crashed without an error output. For this reason the Twentse News Corpus had to be shortened to 81 million words. Because the program directly calculates similarity between words there was no need (and possibility) to compute the true sense of the word. The output is, like all the approaches above, a similarity matrix. The approach from Mikolov et al. (2013) will be abbreviated to Word2Vec.

Determine dimensionality

Matlab[6] is used to perform the clustering of the data. The data is supplied in form of a symmetric matrix that is generated by the algorithms discussed above. Before the data can be clustered, it needs to be transformed from a two dimensional similarity matrix to an *n*-dimensional space, because the clustering algorithms in Matlab are not able to cluster matrices. This was done by applying multidimensional scaling (MDS). These techniques allows a researcher to convert similarity

[5] http://radimrehurek.com/gensim/index.html
[6] http://www.mathworks.nl/products/matlab/

(or dissimilarity) measures to a multidimensional space (Borg & Groenen, 2005) and provide the ability to literally "look" and explore the data and its structure visually. Before transforming the symmetric matrix to a *n*-dimensional plot the following question has to be answered: "How many dimensions does fit the given data?". To measure how a given configuration (i.e., n-points in a t-dimensional space) fits the data the stress measurement will be used. Kruskal (1964) defines stress to be: "*a 'residual sum of squares', it is positive, and the smaller the better.*" and adds that it can conveniently be expressed as a percentage.

$$stress = \sqrt{\frac{\sum_{i<j}\left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}}$$

Where d_{ij} is the distance between the coordinate and the its dissimilarity and $\hat{d}_{ij}$ is defined as "*merely a monotone sequence of numbers, chosen as "nearly equal" to the d_{ij}, as possible, which we use as a reference to measure the non-monotonicity of the numbers*" (Kruskal, 1964, p7). This stress measurement tells how well the dimensional configuration fits the data. Kruskal (1964) identified the following stress scores and their related assessed fitness.

Stress	Assessment of fit
20%	Poor
10%	Fair
5%	Good
2.5%	Excellent
0%	"Perfect"

Table 3 Stress Goodness of fit

By "perfect" is meant a relationship between dissimilarities and the distances that is equal except for some small discrepancies. (Kruskal, 1964; Borg & Groenen, 2005). A commonly used method is to calculate the stress for each configuration and to plot this against the used dimensions. The stress decreases as the number of dimensions increases. When plotted an scree plot is formed. A scree plot displays a decreasing line, this line can be represented by an elbow. The point of the elbow is interesting, because at this point the stress will not decrease significantly when more dimensions are added. Galbraith, Moustaki, Bartholomew & Steele, 2002). This method is called the "*elbow-method*". Although the elbow-method is a very subjective approach some scholars state that the approach often works very well (Galbraith, Moustaki, Bartholomew and Steele, 2002). This elbow approach will also be applied in this research. The function PROXSCAL in SPSS[7] will be applied to calculate the normalized raw stress for each configuration per algorithm.

[7] http://www.ibm.com/software/analytics/spss/

Determine cluster count

Before clustering can commence, the amount of clusters need to be determined. This will be done with the silhouette technique. The silhouette technique can be used to calculate how well each object lies with its cluster and how far separated the clusters are from each other (Rousseeuw, 1986). The output of the techniques gives an indication to how well the cluster count fits the data. This output is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Where $a(i)$ is the average dissimilarity of i with all other data within the same cluster and $b(i)$ is the lowest average dissimilarity of i to any other cluster which i is not a member. The ideal value of $s(i)$ is as close as possible to 1. A high value indicates the object is well-matched to its own cluster, and not to that of other clusters. The average value over all data in the clusters tells how well the objects have been clustered. A high value tells that the clustering is fitting the data, when silhouette technique returns a low value it can either mean there are too few or too many clusters.

Clustering

Now the dimensionality and amount of clusters is known it is possible to cluster the words. The result from the similarity calculation will be used as input. This is a symmetric matrix and cannot be directed be clustered, because the cluster algothihms need a *n*-dimensional plot. First, all the data points needs to be transformed so that the words can be plotted in a *n*-dimensional plot. This will be done by the *cmdscale* function in Matlab. This function gives the possibility to transform the data in a certain *n*-dimensional space. In this case the *n*-dimensional space that is determined by SPSS.

When the data is in a *n*-dimensional space it is possible to start applying clustering techniques. The techniques agglomerative and K-means clustering will be applied. These are the most commonly used clustering techniques (Manning & Schütze, 1999). As stated in 2.3.3, there is no clear sign which of these is better. To rule out that they can influence the results both will be tested. After testing both techniques one will be chosen to continue with.

Evaluation

During the external evaluation the results from the clustering will be benchmarked against existing knowledge. Most of the time the existing knowledge serves as a golden standard for evaluation. Existing knowledge can be the structure of a ontology or clustering results done by human informants. The evaluation measures that will be discussed in this subsection describe how to measure the results compared to that of existing knowledge.

Purity

The purity measurement calculates how accurate the algorithm assigned the object to the correct clusters and based on this information determines the accuracy. In order to calculate purity the method counts the number of correctly assigned objects and will be divided by *N*, where *N* is the total number of objects.

$$\frac{\sum_{i=1}^{N} C_i}{N}$$

The result of the measurements ranges from zero to one, where zero represents bad clustering and one perfect clustering. A disadvantage is that high purity values is easy to achieve. When there are large amount of clusters the purity value will rise, because there are fewer objects in each cluster.

A purity value of one is achieved when each object has its own cluster. Thus, the method cannot be used to say something about the number of clusters and its quality (Manning, Raghavan & Schütze, 2009). The silhouette functions forms a backup to determine the '*right*' number of clusters.

Normalized mutual information

With the normalized mutual information (NMI) it is possible to say something about the cluster count and their quality. NMI measures how many of the objects have been correctly or incorrectly classified (Manning, Raghavan & Schütze, 2009). It is possible to compare cluster results with different cluster counts, because NMI is normalized, thus the outcome of NMI will always be between zero and one. The formula of NMI is:

$$NMI(W, C) = \frac{I(W; C)}{[H(W) + H(C)]/2}$$

Where $I(W; C)$ is:

$$I(W; C) = \sum_{k} \sum_{j} P(w_k \cap c_j)\, log \frac{P(w_k \cap c_j)}{P(w_k)P(c_j)}$$

And H is:

$$H(c) = -\sum_{k} p(w_k) \log P(w_k)$$

The part of $I(W; C)$ in the equation measures the amount of information the clustering represents compared to that of existing knowledge. Like purity this measurement will not penalize the increase of clusters, thus leading to an increase of cluster quality. The denominator helps to fix this problem,

because when the cluster count increases the entropy will also increase (Manning, Raghavan & Schütze, 2009).

Rand index

Another method to measure the external validity of the clusters is called the Rand-index. This measures the percentage of objects that are clustered correctly (Manning, Raghavan & Schütze, 2009).

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

True positive (TP) is when two similar objects are assigned to the same cluster, a true negative (TN) when two dissimilar objects are assigned to different clusters. Then there are two type of errors, namely: false positive (FP) when two dissimilar objects are assigned to the same cluster or a false negative (FN) when two similar objects are assigned to different clusters. Basically the measurement is a type of accuracy measure. The measurement gives equal weights to false positives and false negatives.

F-measure

The F-measure is able to give weights to these errors, because separating similar objects is sometimes worse, compared to combining dissimilar objects to the same cluster (Manning, Raghavan & Schütze, 2009). The measurement uses precision and recall in order to calculate a final score. Precision is the amount of objects that are correctly assigned to their cluster, where recall looks at the amount of documents that belong to a cluster and were actually assigned to it. These are calculated in the following way:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

The traditional measurement, where no weight was given, can be seen as a harmonic mean between precision and recall. Thus in the following formula β will have to value of 1.

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

But it is also possible to give extra weight to penalize false negatives more strongly than false positives, by assigning a particular value to β.

Applying evaluation approaches to the research

NMI or the silhouette technique will only be applied when the number of clusters is unknown. The Silhouette technique will be used, because it is a part of Matlab. Thus, with the first and third word collection where the cluster count have already been set there is no need to find out how many clusters are in the data. Here the task of the algorithm is to replicate the given clusters as close as possible. To evaluate the results the measurements that will be used are: (1) purity, and (2) F-measure. These methods are also frequently used by other scholars in order to measure the quality of their clusters (Bronda & Mazur, 2012; Agirre & De La Calle, 2003; Jing, Ng & Huang, 2009) and (Martin-Wanton & Berlanga-Lavori, 2012; Matsuo, Sakai, Uchiyama & Ishizuka, 2006; Nasiruddin, 2013; Russo, 2008; Miller, Guinness & Zamanian, 2004; Saha, Mitra & Sarkar, 2008; Ghayoomi, 2012) respectively.

During evaluation the result will be compared to existing knowledge (i.e., how well it can replicate that what was given by human informants). Because there is no consensus about how to evaluate word clustering, several approaches need be tested. These are all related to the approach of word collecting, the most interesting is the use of human informants, because they will also be the judges in the real life situation. As stated in the Word Collection stage, two approaches are created: (1) the informants judge given word clusters, and (2) the informants cluster the words themselves and the application tries to replicate these results. To highlight the ambiguity of word clustering we are going to ask other informants to cluster the words given by one of the earlier groups. This is done to see how much agreement there is between people when it comes to clustering. To measure agreement we are using the Kappa coefficient. Normally the Cohen's Kappa (Cohen, 1960) is used, this measure the agreement between two raters and results in a percentage agreement calculation. In this research the Cohen's Kappa cannot be used, because we want to see the agreement between multiple raters. Therefore, the choice has been made to use the Fleiss Kappa measurement (Fleiss, 1971). The different between Cohen's Kappa en Fleiss Kappa is that the latter can handle multiple raters.

3.3.2. Communication

The findings of the research will be communicated in two ways. First, through the thesis containing detailed description on how the research was performed, results gathered, analysed and discussed. There will also be a verbal defence where the highlights of the research be presented in front of a scientific audience. Emphasis will be on theoretical foundation, research methodology, and in-depth discussion of the results and its implications.

3.4. Summary

In this chapter, the research methodology of this study has been discussed, what design shall be followed and how it is actually implemented and evaluated. This to ensure the research is conducted in a scientific and structured manner. From literature several factors have been identified that could influence the process of clustering. From these factors the method, display in figure 1, has been synthesized. This method has been developed and evaluated according to the design process of Peffers et al. (2008). The actual implementation has also been explained in this chapter and will be followed in the following chapter. Also, the method acts like a template where all steps can be altered. This enabled the possibility to test other configurations.

4. Experimental results

In the previous chapter a process method to perform word clustering has been synthesized. In this chapter this method will be applied in order to find the configuration that gets the best results. This will be done in two cycles. First, elimination of suboptimal configurations. During this step the best word similarity algorithm will be determined and which clustering technique best fits the task. In the second cycle the process is tested with input from human informants and also evaluated by them.

4.1. First cycle – eliminating configurations

Before in-depth analysis of the effect of similarity algorithms on the clustering results the two clustering algorithms will be compared to each other. The initial one hundred, ten word class, input was given and both algorithm were set to give ten clusters as output. The purity technique will be applied to measure how much the clustering algorithms accurately the existing knowledge.

Name	K-means	Agglo
AKC	.73	.71
JNG	.39	.40
LIN	.16	.17
RSK	.21	.17
Word2Vec	.50	.59
WUP	.72	.72

Table 4 Compared algorithms measured purity

In the table above it is visible that the clustering algorithm does not really influence the result. We use the purity to compare the two algorithms to each other. Other scholars (Steinbach, Karypis & Kumar, 2000; Chen, 2005; Feize-Derakhshi & Zafarani, 2012) use the F-measure while comparing, but as will be elaborated on in the discussion we see something peculiar with this method. Except for Resnik (1995) and Mikolov et al. (2013) are the results almost identical. The preference is given to the k-means algorithm and will be used in the following experiments.

The next step is to compare each similarity algorithm to one other, while using the same clustering method. Before applying clustering to each similarity algorithm the dimensionality needs to be determined. The dimensional count and stress value for each algorithm are displayed in the table below.

Name	Dimensions	Stress
AKC	6	.011
JNG	4	.029
LIN	6	.064
RSK	6	.263
Word2Vec	6	.012
WUP	7	.008

Table 5 Determining dimensions by applying the elbow-method

According to Kurskal (1964) it might be expected that the algorithms that score the lowest stress value will score better compared to those with a higher stress value. The dimension number in the table above will be used in during clustering. All similarity matrices are read by Matlab and clustered using the k-means algorithm, as above, the k-means iterations have been set to one hundred. The results of the clustering method were compared to the existing knowledge about the word classes.

Name	Purity	RandIndex	Precision	Recall	F1-Score
AKC	.730	.931	.606	.680	.641
JNG	.390	.857	.276	.353	.310
LIN	.160	.605	.118	.518	.192
RSK	.210	.475	.110	.678	.190
Word2Vec	.500	.880	.373	.467	.415
WUP	.720	.925	.571	.711	.634

Table 6 results word similarity algorithms

In the table above the results of the different similarity algorithms presented. It is visible that the algorithms of Altintas, Karsligil and Coskun (2005) – AKC and Wu and Palmer (1994) – WUP perform the best. The purity of these two algorithms is 73% and 72%, respectively. Remarkable is the performance of Resnik (1995) – RSK, Jiang and Conrath (1997) – JNG and Lin (1998) – LIN. They all score accuracy under forty percent. These algorithms all belong to the node-based approach that use information content during their similarity calculations. Also, these algorithms differ in output. Their output is between zero and infinity where the output of the edge-based and corpus-based approach all are between zero and one. The performance of the technique introduced by Mikolov et al. (2013) – Word2Vec is reasonable and scoring an accuracy of 50% placing its performance between that the edge- and node-based approaches.

We choose the algorithm of Wu and Palmer (1994) and will be used in the second cycle as the primary algorithm to calculate word similarity. In addition to the Wu and Palmer (1994) approach will also the approach of Mikolov et al. (2013) be used, because of its significant larger vocabulary size compared to that of Wu and Palmer (1994). We predict that the approach of Mikolov et al. (2013) will recognize more words, compared to approaches that use an ontology like WordNet or Cornetto. The Cornetto database, used by Wu and Palmer (1994), contains 92K lemmas (70K nouns, 9K verbs, 12K adjectives and 73 adverbs) corresponding to 118K word meanings[8] and is expected to grow in the future. As similar to the database of WordNet. In 1993 it contained 96K words organized into 70.000 word meanings, or synsets (Miller et al., 1993). At the moment WordNet covers over 155.000 words organized into 117.000 synsets. The approach of Mikolov et al. (2013) trains its vocabulary on a given corpus. When trained on the Twentse News Corpus the vocabulary has the following size:

[8] Cornetto Lexical Database Documentation, Cornetto Deliverable D-16, Version 7, Januari, 2009

761,000 and 1.85 million when fed corpus with size 81 million and 327 million words, respectively. The advantage to train the vocabulary on a corpus is that it can learn domain specific words and if we train on a larger corpus it can contain the entire vocabulary of a native speaker.

4.2. Second cycle – human informants

Human informants will be used two times to supply words, thus replicating a more realistic collection of words and as judges of the results. First, the results will be discussed where the informants were asked to evaluate and where needed to correct the clustering results. Afterwards the results of the second evaluation, where the informants were asked to cluster the words before the algorithm does the cluster. Here the algorithm needs to replicate the results given as best as possible.

4.2.1 First evaluation

In the first experiment using human informants the objective is to obtain data from and evaluate the results by the subjects. The results of this experiment can be found in appendix B. A group of fifteen master TBK/BIT[9] students from the University of Twente were asked the following question: "Welke activiteiten onderneem je in het weekend?" (Which activities do you embark on during the weekend?). Each student entered their answers to a web-application, which later was used to display the cluster results and ability to give (if needed) corrections. The fifteen students entered 99 words out of which 57 were unique. These words were supplied to the algorithms WUP and Word2Vec to see which one recognized the most words. WUP did not recognize 38 words (66%). After stemming six more words were recognized. After checking the remaining words the following points came to attention: (1) illegal compound words (i.e., familiebezoek [visiting the family]), (2) English words that are used in the Dutch language (i.e., gamen) and (3) misspelled words (i.e, whisky). Spelling checkers and stemming techniques can be applied to increase the recogzined word count, but this can have an effect on the sense of the words. The word '*internetten*' (browsing the internet) is stemmed to '*internet*'. It went from a activity to an object. Because of the low recognition by WUP the Word2Vec algorithm was applied. Word2Vec recognized all the words, except for one (filmkijken). The choice was made to continue with Word2Vec, otherwise there would be too much loss.

Before clustering the stress was calculated in order to determine how many dimensions should be used. The plots used during this experiment are presented in appendix B. The earlier discussed elbow-method has been applied that suggested to six dimensions. At this point the stress is .0196, according to Kruskal (1964) this a near perfect fit between the data and dimensionality. The second step is the internal cluster analysis, performed using the silhouette method in Matlab. After

[9] Technical Business Administration (TBK), Business and Information Technology (BIT)

observation the choice was made for eight clusters, which corresponds to a silhouette value of .426. Although ten and eighteen have higher silhouette values, .465 and .527, respectively.

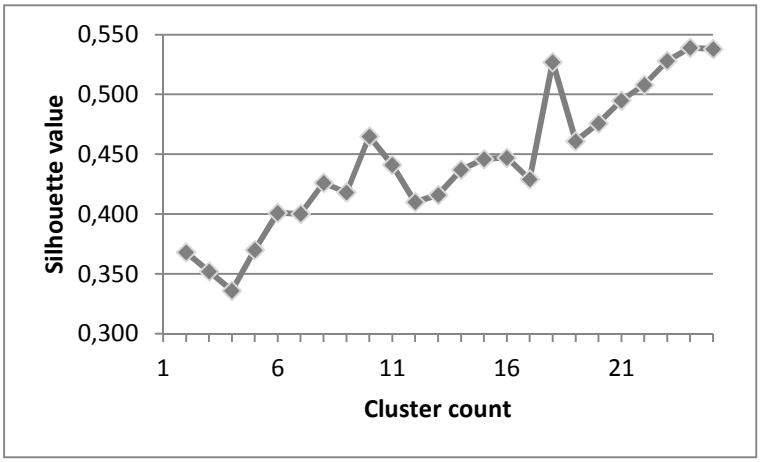


Figure 4 Relation between cluster count and silhouette value

In the figure above the relation between cluster count and silhouette value is displayed. The more clusters are added the better the silhouette value becomes, this is because the tightness of the cluster will decrease and the separation will likely increase. The choice was made to have a lower cluster count that still has a reasonable silhouette value. Also, because the main purpose of clustering is to reduce object count and get a general representation of the data. Choosing a high cluster count increase the change of over fitting the data (e.g., when choosing as much clusters as there are object the purity of the algorithm would be 1). Finally the data has been clustered using the k-means algorithm. After clustering the words all the clusters were presented to the students and they were given the possibility to move words that, according to them, did not belong to the correct cluster. In total seventeen corrections were made. After processing the corrections seven classes remained. The word class is used to indicate the correct state, where clusters are generated by the algorithm. With these two variables it was possible to calculate the purity of the sample which is .80, thus 80% of the words were clustered correctly.

4.2.2. Second evaluation

In the second experiment a group of students were asked to answer the same question as posed in the first experiment, but after answering the question they were also asked to cluster these results. The goal of the second experiment was to see how the algorithm can replicate the existing knowledge, or predefined clusters. In the table below the results from these experiments are displayed. In-depth results can be found in appendix C.

Group	Students	Words	Unique	Purity
1	4	50	39	.378
2	4	40	27	.282
3	6	47	20	.350

Table 7 Second experiment

4.3. Summary

In this chapter several configurations were eliminated to find the optimal configuration that best performs the task of word clustering. The process method serves as a good platform to find this configuration in each of it steps options can be added or removed to test different configurations. Further findings will be discussed in the next chapter.

5. Discussion

In this chapter the results found during the experiments will be discussed. Also secondary findings will be discussed, these are findings that are not directly related to the research question, but resulted during the experiments and are interesting to take a further look at.

Information content

After examine the approach of Resnik (1995), Jiang and Conrath (1997) and Lin (1998) one flaw of the information content approach became visible. According to Resnik (1995) the basic intuition behind information content is the probability of concepts. The more probable the appearance of a concept is the less information the concepts carries, thus frequent words are less informative compared to infrequent words. When applying the approach to a tree structure one should expect that the leaves of the tree carry more information compared to the branches and the root. Thus, the information content of words at the bottom of the tree should be higher. This has been tested with the words "bicycle" and "car". Here it becomes evident that the content information does not represent the tree structure. Words like *transportmiddel* (means of transportation) and *motorrijtuig* (motorized vehicle) possess a higher information content value compared to *fiets* (bicylce) and *auto* (car). One could expect these results seeing the usage of these words in sources like the Twentse NewsCorpus. Here more specific terms will be used, meaning that more abstract terms will not appear often in the corpus. The implication is that word share a common ancestor like *transportmiddel* will be more similar to each other, compared to words that share an ancestor like *vehikel* (vehicle), because this ancestor will have a high lower IC value compared to *transportmiddel*. Even with this defect it is remarkable that the algorithm by Jiang and Conrath (1997) performs better compared to the other information content approaches (Resnik, 1995; Lin, 1998).

Subjective-ness in cluster evaluation

After the first experiment using human informants, accuracy of the algorithm was 80%, but there are two remarks about this value. First, that when no corrections were given the purity would be 1. Meaning, a small number of corrections would have a small effect on the purity. Second, the people were not able to create new clusters and place the words there. The algorithm looks at which correction was given most often and places the word in that cluster.

The second experiment with human informants was designed to catch the design flaw, namely; of corrections and ability to determine cluster count. The human informants were asked to cluster the answers they have given, before the algorithm would do this. The task for the algorithms was to replicate the results from the human informants as best as possible. Directly it was visible, see table 7, that the purity is far lower compared to earlier performed experiments. Here the bias of clustering

accuracy becomes visible. Human informants are the only ones that can truly judge the results, but replicating their results is extremely difficult. Clustering results depend a lot on the subjectivity of the informants. When presented with pre-clustered results they faster agree with what is presented. During judging they can play with the sense of words, or imagine certain context that fit the words and then justify that the words belong to the right cluster.

This subjectivity becomes visible when a list of words is clustered by different persons. To demonstrate this three people were asked to cluster the words given by group one from the earlier discussed experiment. The table below displays the results:

Clustered By	Clusters	Purity	Cohen's Kappa
Original	6	.378	
Person #1	7	.459	.672
Person #2	11	.297	.475
Person #3	9	.405	.525

Table 8 Clustering results

Same data, different results. This is also one of the problem to accurately measure purity when there are multiple correct answers. There is a big chance that when these results are presented to other informants that they would agree with what is presented to them. When each person's result was compared to that of the original group the agreement between them is, according to Landis and Koch (1977) between moderate and substantial. When all raters are compared to each other with the Fleiss Kappa measure the result is .496. According to Landis and Koch (1977) this the strength of agreement is moderate. Less ambiguous words were all clustered together. For example, there are 6 words that share a main theme 'sports'. All the four persons clustered these words together, they all agreed with the relation these words share. Other more ambiguous words were found in different clusters, in one instance more clusters were used to make a better fit between the clusters and the data. This can be explained by the sense people give to the words or how the words are related to each other due to the personal identification with these words.

Accuracy remains a trivial concept. It becomes clear that human informants need to agree with the results and not trying to replicate a certain clustering result. Although the first experiment has the problem that the lack of corrections could lead to a higher purity value, this can now be explained by the subjective-ness in determining the clusters by these informants. The ability to play with word sense and/or context are likely the reasons that when asked to judge results the purity is significantly higher, compared with trying to replicate the clusters defined by informants.

Dimensional impact

As stated earlier the elbow method was used to determine the n-dimensional space used to transform the similarity matrix. In hindsight we can confirm the finding from (Galbraith, Moustaki, Bartholomew & Steele, 2002) that the elbow-method is a particularly good method to determine the right dimensional count. To see how the different *n*-dimensional space impacts the result the following experiment was performed. One hundred, ten word classes were taken as input and clustered using different dimensions, ranging from two to ten dimensions.

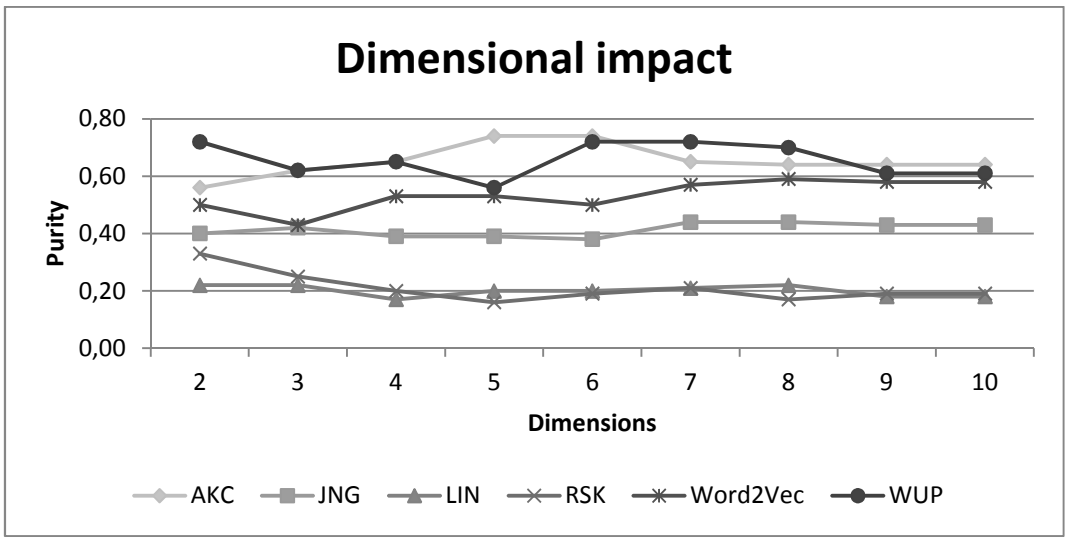


Figure 5 Dimensional impact on clustering results

In the figure above the purity is presented for each dimensional setting. It is clear that choosing the wrong setting can influence the performance of the clustering algorithm. Between the right and wrong n-dimensional space there is a twenty percent difference in accuracy, therefore it is important to make sure we apply the MDS technique to determine which dimensions fits the data best.

Purity vs Rand Index and F-measure

In table 6 there is a strange relationship between purity and F-measure. The purity scores from Lin (1998) and Resnik (1995) differ by five percent from each other, but their F-score differs only by .002. Also according to the F-score the algorithm of Lin (1998) performs better, ever so slightly, compared to that of Resnik (1995). The Rand-index and F-measure both build on the calculation of amount of similarity of objects present in similar or dissimilar clusters. This means that when two similar objects are both in the same dissimilar clusters the method will identify them wrongly. We performed the following experiment to determine the relationship between purity and F-measure. Two clusters were made with each ten unique objects. Each iteration one of the objects was moved from cluster one to cluster two, until cluster one was empty. Each iteration purity, the Rand index and F-measure was calculated and plotted in the diagram below.

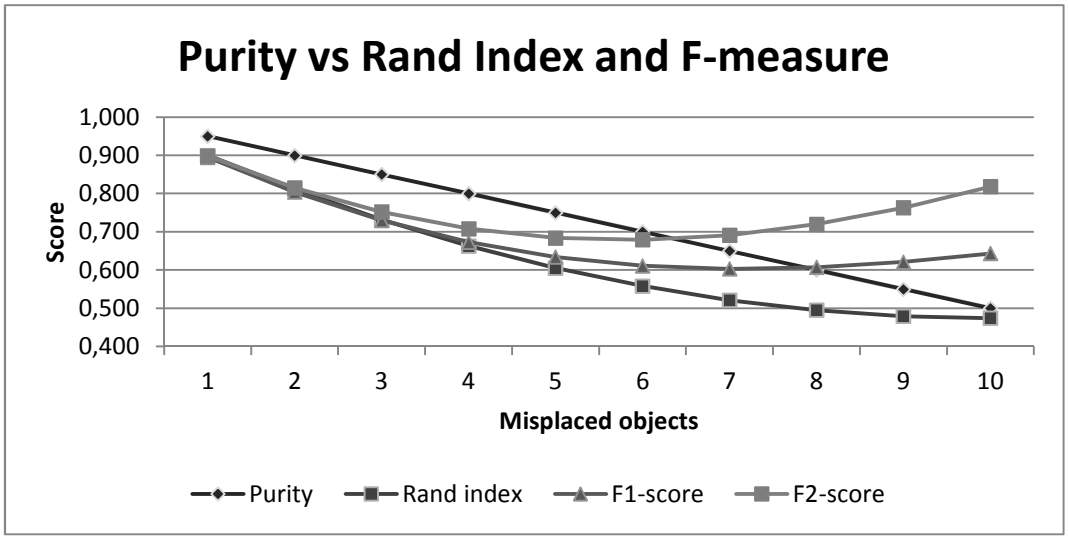


Figure 6 Purity vs Rand Index and F-measure

For the first six iterations all methods penalize misplaced object, but afterwards the F-score begins to rise. The F-measure is the harmonic mean of precision and recall. When these two parameters were examined it became clear that the precision almost follows the same decreasing trend as the Rand Index. Here we provide an example of the problem with recall; the source of the problem is the recall, after hitting .722 at iteration five it slowly climbs back to 1. If we take the following classes $C_1 = \{w_1, w_2\}$ and $C_2 = \{w_3, w_4\}$ and the following cluster $c_1 = \{w_1, w_2, w_3, w_4\}$. And check the true positive condition, similar objects in similar clusters, this is true for sets: $\{w_1, w_2\}$ and $\{w_3, w_4\}$. The method does not see which clusters it is really comparing to each other. Therefore, from now in comparing the various algorithms to each other the purity method will be used, because it measures the accuracy.

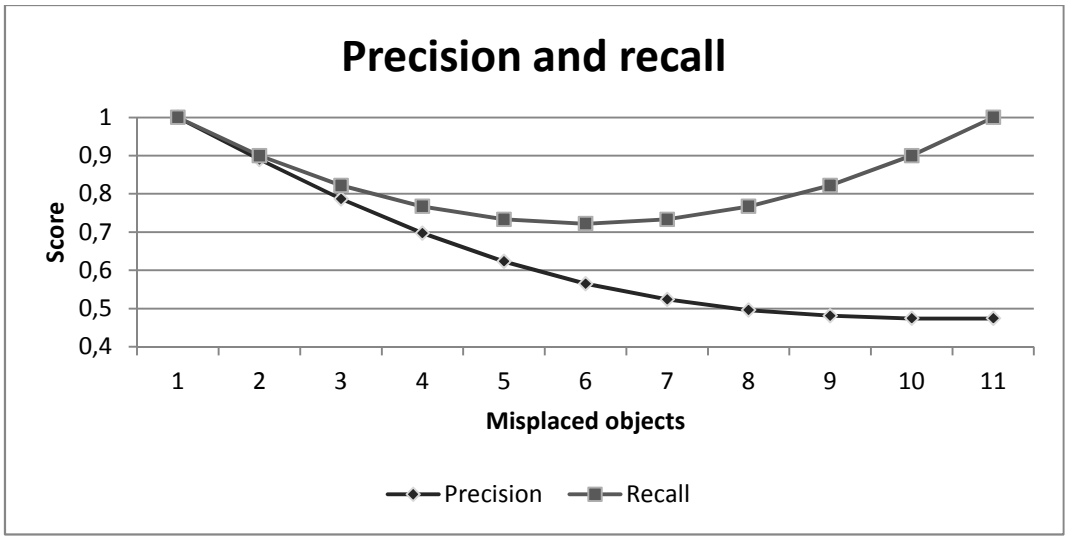


Figure 7 Relation between precision and recall

Recall is being influenced by the number of true positives and false negatives. In this experiment the latter acts as the negative of the true positive, thus the denominator in the recall formula stays

stable. Meaning that the true positive is the only variable that influences the recall. To answer the question why recall displays this U-shape, we think the reason is after 50% of the object have been misplaced the majority of the objects are in the same class and cluster, as explained in the example before the objects are not in the correct cluster. The recall does not see that the cluster and class which it is comparing are different from each other. Knowing this the Rand index and F-measure will not be used to compute cluster accuracy.

Purity deterioration

Increase of word count also influences the accuracy score of the various algorithms. During the experiments different word counts have been used to see how this influences the individual algorithms. As part of the experiment we started with twenty words and add one class containing ten words each iteration until the maximum of one hundred words were been reached. The deterioration of accuracy was as expected. Adding words does influence word sense disambiguation process is seen. In the figure below the deterioration of the different algorithm when increasing the word count.

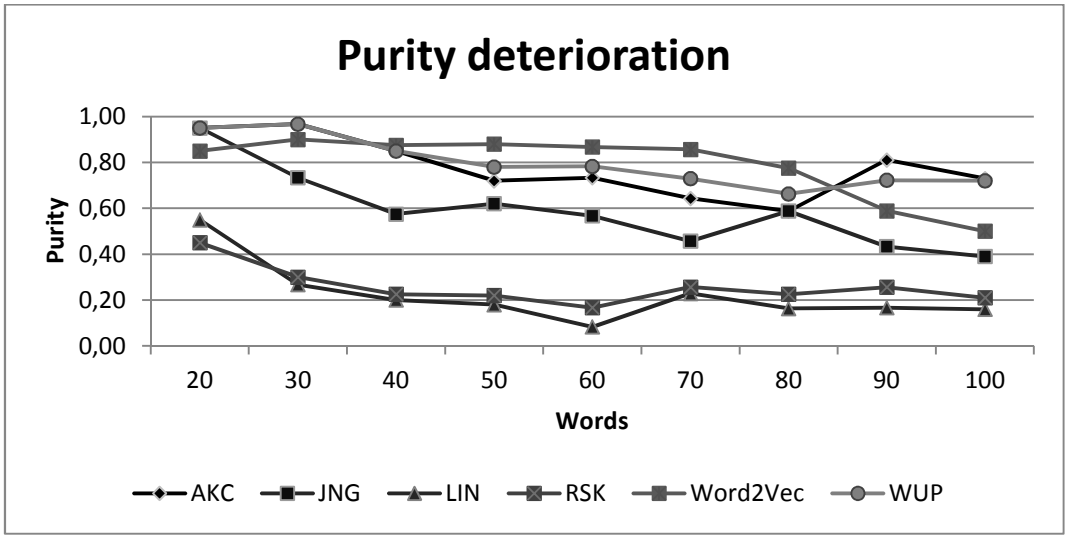


Figure 8 F-score deterioration by word count

Interesting to see here is the performance of the node-based or information content approaches. They are all based on the same underlying approach, but the algorithm from Jiang and Conrath (1997) score reasonable good with smaller word counts. Even almost matching the performance of edge-based approaches when eighty words were used. Still the overall winner are the edge-based approaches. Although, the algorithm from Altintas, Karsligil and Coskun (2005) scores a higher purity score at the one hundred mark . The algorithm from Wu and Palmer (1994) scores on average better compared to that of Altintas, Karsligil and Coskin (2005). Therefore, the algorithm from Wu and Palmer (1994) will be used in the second experiment.

Labelling the clusters

Deepak, Rao and Khemani (2006) posit the following hypothesis: "*The points closest to the cluster center are representative of the cluster*". The N closest words to the centroid represent the cluster. Deepak, Roa and Khemani (2006) call these words "*representative words*". Words that are further removed from the centre are less semantically coherent to the words in the cluster. We tested this during the second experiment. We choose three words that would be the representative words each of the clusters. A smaller number was used, because amount the words we had was also not that large. Even with a small number of representative words, some clusters only exists of representative words. This can mean that the representative words are not really representing the cluster, because of the lack of other words.

The representative words were used to determine a more abstract concept of the cluster and to see if the other words can be explained by this abstract. For most clusters this was applicable, sometimes a cluster would contain words that are definitely wrongly assigned to that cluster, or words that are wrongly assigned to another cluster. When looking at cluster 2 from group 2 in appendix B it is visible that the representative words describe the cluster as one that has something to do with sport. Not surprising that the word 'sporten' (sporting) is also a member of this cluster.

6. Conclusion

This chapter answers the central research question of this study. This question is:

"*Is it possible to reliably cluster language independent individual words in a given communication context?*"

In order to answer the research question, first the sub questions need to be answered. Starting with: "*Which factors influence the clustering process?*". During the literature review the following factors that can influence the clustering result were identified: (1) word similarity algorithm, (2) dimensionality, (3) clustering algorithm, (4) cluster count, and even (5) evaluation approach. These factors were placed in a process method figure 1 shows the process of word clustering.

The information content algorithms Resnik (1995), Jiang and Conrath (1997) and Lin (1998) performed the worst. This was because of a shortcoming in their approach that made their algorithm unsuitable. Their basic intuition of concept probability did not hold up during calculation on the Twentse News Corpus. The performance of Jiang and Conrath (1997) is still remarkable. They use the same underlining approach, but is able to outperform Resnik (1995) and Lin (1998). The edge-based approaches performed the best, but did not hold up their supremacy during experiments with human informants. The problem was with word recognition by the Cornetto database. The ontology did not cover enough words that were given by the informants. Here the approach of Mikolov et al. (2013) showed its strength. By learning on a corpus it became possible to learn a wide range of words from a given language and at the same time calculate word similarity. The benefit of this approach is that the algorithm can learn domain-specific words and is language independent.

During the transformation process the similarity-matrix needed to be transformed to an *n*-dimensional space in order to use clustering techniques. Minimizing stress improves the fit between given data and the dimensional configuration Kruskal (1964). After testing the impact of different *n*-dimensional spaces on the clustering results it became clear that this factor can strongly influence the result. A difference of twenty percentage was measured between dimensional configuration that fit and does not fit the given data. The elbow-method posited by Galbraith, Moustaki, Bartholomew and Steele (2002) is a good method to determine the right dimensionality.

Different clustering techniques does not influence the results. The tested techniques Agglomerative clustering and K-means performed almost identical to each other.

To determine the cluster count the Silhouette function of Matlab was used. This function gives a representation of cluster cohesion and separation. A higher Silhouette value implies that objects

belong to their rightful cluster and not to other clusters. This step is needed to determine how many clusters exists in the given data. As we will see that cluster count does not directly influence the result, because of the subjective-ness of human informants. This step is needed in order to cluster the data.

During evaluation something remarkable happened. Two experiments using human informants were performed. In the first experiment the informants needed to judge the given clusters and correct then where needed, during the second experiment they were asked to cluster the answers by themselves and the clustering algorithm's task was to replicate these results as best as possible. In the first experiment the purity was around 80%, but with the second experiment this was around 30%. The sudden drop in purity can be explained by the subjective-ness of the informants. Humans are able to see or agree with the different configurations of clusters, thus there is no single right answer. To prove this, three other people were asked to cluster a set of words supplied by one of the groups during the second experiment. Although their agreement was between moderate and substantial (Landis & Koch, 1977), this clustering resulted in three totally different results. Not only were the words clustered differently, but also different cluster counts were used. This makes it extremely difficult to determine what really is the correct state. Therefore, one should use human informants be used to judge the given clusters and correct them where needed.

The second sub question is: "*How can the similarity between words be calculated?*". The edge- and node-based approaches are dependent on word sense disambiguation. The Cornetto ontology is only able to calculate similarity when for each of the words the true sense of the words has been determined. Incorrect disambiguation can result in an incorrect assignment of clusters. Here the algorithm of Mikolov et al. (2013) has again the advantage that it can calculate word similarity without having to use the true sense, but this means there is the possibility that words can have partial membership to clusters. For example, with the words: football, match and lighter. The word 'match' can be plotted between football and lighter. The K-means algorithm would randomly assign the word to one of the clusters. Because we do not know the true sense we cannot say it does not belong to the other cluster. Partial word membership can solve this problem.

Now to answer the central research question. This research demonstrates multiple ways in which the results of brainstorming sessions can be clustered. Testing and eliminating configurations results in the configuration where the word similarity algorithms of Mikolov et al. (2013) give the best result. The algorithm has the advantage that it is language independent and can learn domain specific words. Also, when trained on a large corpus the vocabulary covers almost the entire language of the

native speaker. Thus, when trained on a corpus related to the brainstorm session the algorithm can optimize itself to perform as best as possible to the given situation and adjust to other situations.

6.1. Limitations

In order to reduce complexity of the clustering process the input of the clustering has been limited to single words. Results from a typical brainstorm session can exist: (1) single words, (2) compound words, and/or (3) small sentences. Especially in the Dutch language this forms an issue. At the moment the compound word: "*zwarte cat*" (i.e. black cat). Will be processed as two words, where the first word can join a colour cluster and the latter an animal cluster. The true meaning of the compound word can be something entirely different, like an entity of something.

According to Peffers et al. (2008) multiple iterations of the design process are possible to further improve the artifact. In this research the process method (i.e. artifact) was only tested once without going through a redesign cycle. The method fits the research approach, but the clustering works when the input consists of single words. Changes in the input will definitely influence the design of the artifact.

The algorithm of Mikolov et al. (2013) was only trained and tested with the Twentse News Corpus. According to Deepak, Delip and Deepak (2006) there exists a corpus bias. The semantic relationship between words is determined by the corpus, but can differ between corpora.

During the selection of words an flaw in the Cornetto database was discovered. Because Cornetto, like WordNet, are ontologies all the words in the database must have the same root. In WordNet this is '*entity*' and in Cornetto this is '*iets*'. For example, the words: '*handelen*', '*regelen*', '*maken*' all don't have as root the word 'iets'. If this root is missing there is a possibility that words are not fully connected to each other. Thus, calculating word similarity is not always a possibility.

6.2. Future research

Because of the ambiguity that exists in language it is important that more research needs to be performed on the evaluation of word clusters. As seen in this research multiple methods give very different results. As humans are the only one that can properly judge the results they have be incorporated into the process, but the utilisation of humans can differ. Further in-depth study as to why informants so easily agree with presented clusters can shed light on this issue.

This research applied a hard clustering technique, but there are also soft clustering techniques. These allow objects to have a membership to more than one cluster. This disjunctive clustering can also be

applied to give highly ambiguous words the possibility to become members of multiple clusters. A method that can be used is the Expectation–maximization (EM) algorithm.

As stated in the limitations, this research only focused on single words. Future study should look at compound-words and (small) sentences. When approaching small text possibly named entity recognition can also be taken into account. As with the example with the 'black cat' this could possibly be something like a restaurant of bar, thus giving the entity a totally different sense compared to a black animal.

Larger word input should test the accuracy of the algorithm. At the moment the input size was around 50-60 words. Increasing the input size could mean that the output cluster count will also grow, but when trying to decrease and summarize the data this could also mean that the clusters will contain more words. Deepak, Rao and Khemani (2006) suggested an approach to label the clusters by using representative words. This approach could be helpful when the word count within a cluster increases.

Our proposed approach works independent of the input language through the algorithm of Mikolov et al. (2013), but this has not been tested. Future research should look in to this, to see if different languages could influence the word similarity process. Also within the Dutch language different corpora should be tested to see how this influences the similarity calculations.

Finally, the method itself should be tested for completeness. As stated before, including compound words and small sentences will definitely change the process. Still, even without these changes the process should be examined if the current steps are enough or addition of other steps could increase performance.

Bibliography

Agirre, E., & Lopez De Lacalle , O. (2003). Clustering wordnet word senses. *Conference on Recent Advances on Natural Language*.

Agirre, E., & Rigau, G. (1996). Word sense disambiguation using Conceptual Density. *COLING '96 Proceedings of the 16th conference on Computational linguistics - Volume 1*, 16-22.

Agirre, E., Martínez, D., de Lacalle, O., & Soroa, A. (2006). Two graph-based algorithms for state-of-the-art WSD. *In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 585-593.

Agirre, E., Rigau, G., Padró, L., & Atserias, J. (2000). Combining Supervised and Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. *Computers and the Humanities*, 103-108.

Altintas, E., Karsligil, E., & Coskun, V. (2005). A New Semantic Similarity Measure Evaluated In Word Sense Disambiguation. *Proceedings of the 15th NODALIDA Conference(Joensuu, May 20-21,2005)*, 8-11.

Bordag, S. (2006). Word Sense Induction. *Triplet-Based Clustering and Automatic Evaluation*.

Broda, B., & Mazur, W. (2012). Evaluation of clustering algorithms for word sense disambiguation. *International Journal of Data Analysis Techniques and Strategies Volume 4 Issue 3*, 219-236.

Brody, S., & Lapata, M. (2009). Bayesian word sense induction. *In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 103-111.

Chen, J. (2005). Comparison of Clustering Algorithms and Its Application to Document Clustering. Princeton University.

Cohen, J. (1960, April). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement, 20*, 37-46.

Couvreur, C. (1997). The EM algorithm: A guided tour. *In Computer Intensive Methods in Control and Signal Processing*, 209-222.

Cowie, J., Guthrie, J., & Guthrie, L. (1992). Lexical disambiguation using simulated annealing. *COLING '92 Proceedings of the 14th conference on Computational linguistics - Volume 1*, 359-365.

de Vaus, D. A. (2001). *Research Design in Social Research.* London: SAGE Publications Ltd.

Deepak, P., Rao, D., & Khemani, D. (2006). Building clusters of related words: an unsupervised approach. *In PRICAI 2006: Trends in Artificial Intelligence*, 474-483.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 96*, 226-231.

Feizi-Derakhshi, M., & Zafarani, E. (2012). Review and Comparison between Clustering Algorithms with Duplicate Entities Detection Purpose. *International Journal of Computer Science & Emerging Technologies, 3*(3).

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*(5), 378–382.

Gale, W., Church, K. W., & Yarowsky, D. (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. *ACL '92 Proceedings of the 30th annual meeting on Association for Computational Linguistics*, 249-256.

Girotra, K., Terwiesch, C., & Ulrich, K. T. (2010). Idea Generation and the Quality of the Best Idea. *Management Science, 56*(4), 591–605.

Han, J., & Kamber, M. (sd). *Data Mining: Concepts and Techniques.* Morgan Kaufmann; 3 edition (July 6, 2011).

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly archive, 28*(1), 75-105.

Ide, N., & Véronis, J. (1998). Word sense disambiguation: The state of the art. *Computational Linguistics, 24*, 1-40.

Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based On Corpus Statistics and Lexical Taxonomy. *International Conference Research on Computational Linguistics*.

Jing, L., Ng, M. K., & Huang, J. Z. (2010). Knowledge-based vector space model for text clustering. *Knowledge and Information Systems Volume 25, Issue 1* , 35-55.

Klapaftis, I., & Manandhar, S. (2010). Word sense induction & disambiguation using hierarchical random graphs. *In Proceedings of the 2010 conference on empirical methods in natural language processing (pp. 745-755)*, 745-755.

Kolte, S. G., & Bhirud, S. G. (2009). Exploiting links in WordNet hierarchy for word sense disambiguation of nouns. *ICAC3 '09 Proceedings of the International Conference on Advances in Computing, Communication and Control*, 20-25.

Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika, 29*(1), 1-27.

Landis, J., & Koch, G. (1977). The measurement of observer agreement for categorical data. *biometrics*, 159-174.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceeding SIGDOC '86 Proceedings of the 5th annual international conference on Systems documentation*, 24-26.

Lin, D. (1998). An Information-Theoretic Definition of Similarity. *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 296-304.

Lin, D. (1998). Automatic retrieval and clustering of similar words. *In Proceedings of the 17th international conference on Computational linguistics, 2*, 768-774.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *An Introduction to Information Retrieval.* New York, NY, USA: Cambridge University Press.

Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press.

Martín Wanton, T., & Berlanga-LLavori, R. (2012). A clustering-based Approach for Unsupervised Word Sense Disambiguation. *Procesamientto del Lenguaje Natural, Revista n49*, 49-56.

Masood, G. (2012). Word clustering for Persian statistical parsing. *Advances in Natural Language Processing*, 126-137.

Matsuo, Y., Sakaki, T., Uchiyama, K., & Ishizuka, M. (2006). Graph-based word clustering using a web search engine. *EMNLP '06 Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 542-550.

Mihalcea, R. (2007). Using Wikipedia for Automatic Word Sense Disambiguation. *North American Chapter of the Association for Computational Linguistics*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *CoRR, abs/1301.3781*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography, 3*(4), 235-244.

Miller, G., & Charles, W. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes, 6*(1), 1-28.

Miller, S., Guinness, J., & Alex, Z. (2004). Name Tagging with Word Clusters and Discriminative Training. *Proceedings of HLT*, 337-342.

Nasiruddin, M. (2013). A State of the Art of Word Sense Induction - A Way Towards Word Sense Disambiguation for Under-Resourced Languages. *TALN-RÉCITAL 2013, Les Sables d'Olonne, France, 2013*, 192-205.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 69.

Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. *ACL '96 Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 40-47.

Osborn, A. F. (1953). *Applied Imagination: Principles and Procedures of Creative Problem Solving.* New York: Charles Scribner's Sons.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems archive, 24*(3), 45-77.

Purandare, A., & Pedersen, T. (2004). Word sense discrimination by clustering contexts in vector and similarity spaces. *In Proceedings of the Conference on Computational Natural Language Learning, 72*.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, 448-453.

Russo, V. (2008). Clustering and classification in Information Retrieval: from standard techniques towards the state of the art. *Technical Report TR-9-2008 – SoLCo Project*.

Saha, S. K., Mitra, P., & Sarkar, S. (2008). Word Clustering and Word Selection Based Feature Reduction for MaxEnt Based Hindi NER. *The Association for Computer Linguistics*, 488-495.

Schutze, H. (1992). Dimensions of meaning. *Supercomputing'92*, 787-796.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics - Special issue on word sense disambiguation Volume 24 Issue 1*, 97-123.

Sebti, A., & Barfroush, A. A. (2008). A New Word Sense Similarity Measure in WordNet. *Computer Science and Information Technology*, 369-373.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *In KDD workshop on text mining, 400*(1), 525-526.

Tsatsaronis, G., Vazirgiannis, M., & Androutsopoulos, I. (2007). Word sense disambiguation with spreading activation networks generated from thesauri. *IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence*, 1725-1730.

Van de Cruys, T. (2010). Mining for Meaning. *The Extraction of Lexicosemantic Knowledge from Text*.

Velldal, E. (2005). A fuzzy clustering approach to word sense discrimination. *In Proceedings of the 7th International conference on Terminology and Knowledge Engineering*.

Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language, 18*(3), 223-252.

Veronis, J., & Ide, N. M. (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. *COLING '90 Proceedings of the 13th conference on Computational linguistics - Volume 2*, 389-394.

Voorhees , E. M. (1993). Using WordNetTM to Disambiguate Word Senses for Text Retrieval. *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Infofmation Retrieval*, 171-180.

Vossen, P., Hofmann, K., Rijke, M. d., Sang, E. T., & Deschacht, K. (2007). The Cornetto Database: Architecture and User-Scenarios.

Widdows, D., & Dorow, B. (2002). A graph model for unsupervised lexical acquisition. *In Proceedings of the 19th international conference on Computational linguistics, 1*, 1-7.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. *Proceeding ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133-138.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. *COLING '92 Proceedings of the 14th conference on Computational linguistics - Volume 2*, 454-460.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *ACL '95 Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 189-196.

Appendix A – Word List

Dieren	Voertuigen	Instrumenten	Groeten	Weer
olifant	fiets	viool	aardappel	regen
neushoorn	rolstoel	cello	andijvie	wind
aap	step	contrabas	avocado	tornado
haai	skeeler	ukelele	prei	moesson
goudvis	luchtballon	harp	broccoli	temperatuur
nijlpaard	auto	blokfluit	paprika	luchtvochtigheid
tijger	vrachtwagen	tuba	erwt	neerslag
spin	scooter	doedelzak	framboos	onweer
slang	slee	bugel	komkommer	zicht
vlinder	trein	accordeon	knoflook	bewolking

Elektronica	Lichaam	Bouw materiaal	Keukengerei	Meubel
monitor	teen	schroevendraaier	steelpan	bank
modem	voet	boor	wok	tafel
computer	been	bijl	braadpan	stoel
versterker	arm	guts	fluitketel	kast
televisie	hand	koevoet	hogedrukpan	kruk
printer	vinger	krabber	koekenpan	sofa
radio	nek	mes	snelkookpan	tuinstoel
toetsenbord	schouder	plamuurmes	pan	altaar
klok	hoofd	priem	stoofpan	slaapbank
telefoon	oog	vijl	stoompan	boekenkast

Appendix B – First experiment

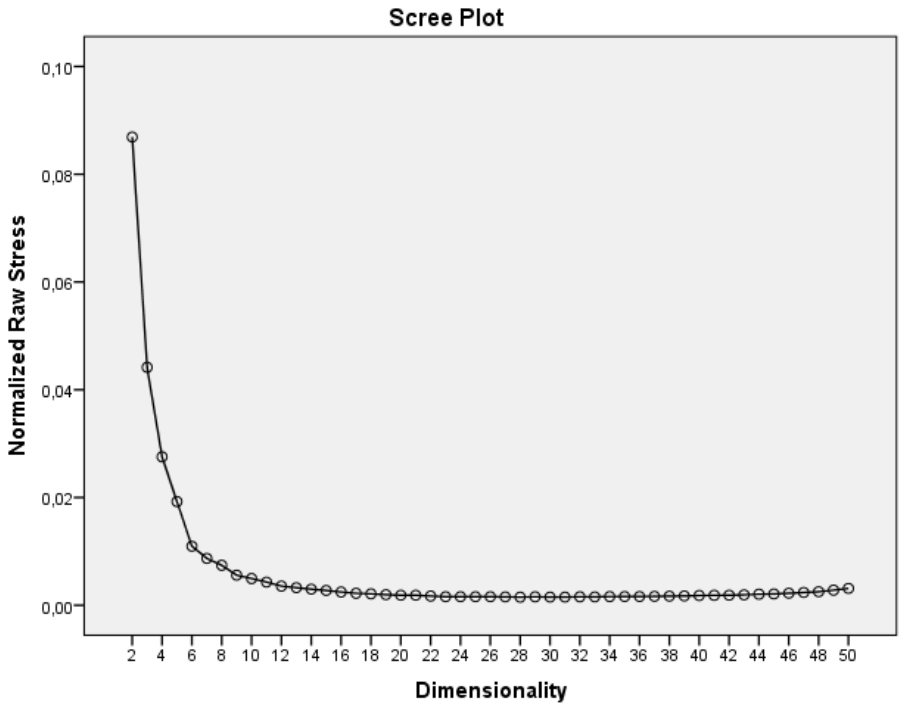
A group of fifteen master students from the University of Twente were asked the following question: “Welke activiteiten onderneem je in het weekend?” (Which activities do you embark on during the weekend?). Each student entered their answers to an web-application, which later will be used to display the cluster results and ability to give (if needed) corrections. The fifteen student entered 99 words from which 57 were unique.

Recognition

These words were supplied to the algorithms Word2Vec and WUP to see which one recognized the most words. WUP did not recognize 38 words (66%), where Word2Vec recognized all the words, except for one (filmkijken). The choice was made to continue with Word2Vec, otherwise there would be too much loss. The following words were not recognized by the WUP algorithm:

bierdrinken, bijslapen, borrelen, coachen, doen, filmkijken, films, fotograferen, gamen, gezellig, hardlopen, internetten, joggen, kijken, koffiedrinken, koken, ontspannen, plannen, reizen, relaxen, series, slapen, spelletjes, sporten, stappen, stedentrip, studeren, tennissen, uitgaan, uitrusten, uitslapen, voetballen, wandelen, wassen, werken, whiskey, winkelen, zuipen

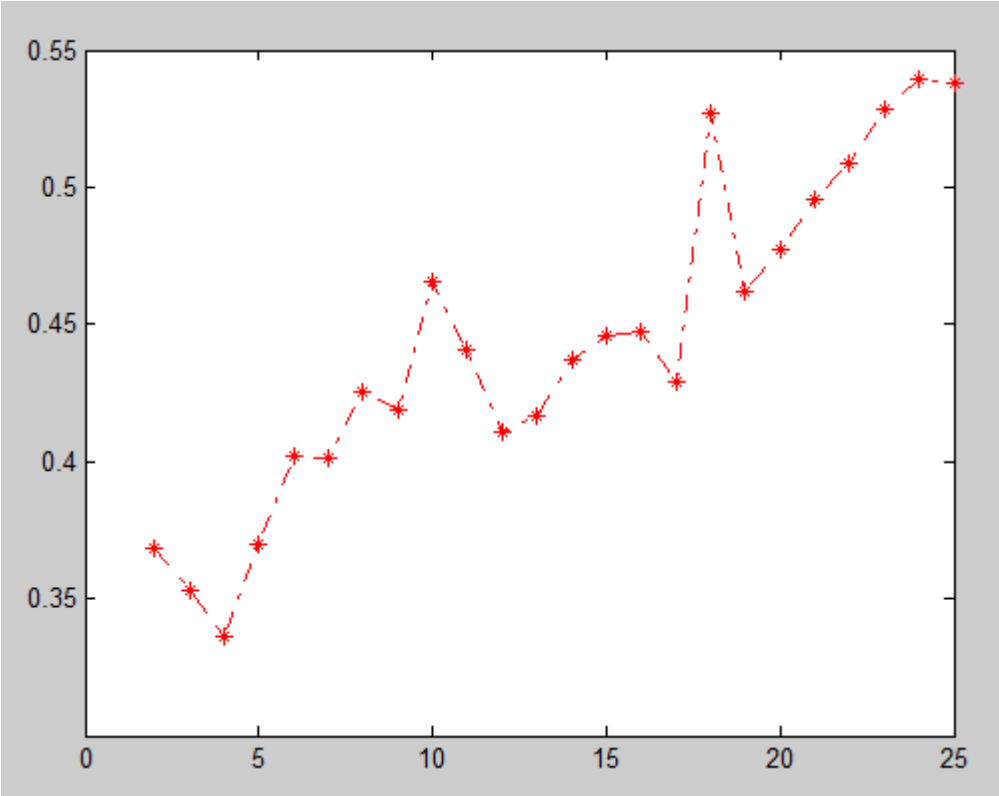
Dimensionality



Figuur 1 Stress vs dimensionality

Before clustering the stress was calculated in order to determine how many dimensions should be used. The earlier discussed elbow-method has been applied and give preference to six dimensions. At this point the stress is .0196.

Determining cluster count



Figuur 2 Silhouette value vs cluster count

The second step is the internal cluster analysis, performed using the silhouette method in Matlab. After observation the choice was made for eight clusters, which corresponds to a silhouette value of .426. Although ten and eighteen have higher silhouette values, .465 and .527, respectively. The choice was made to have a lower cluster count that still has a reasonable silhouette value.

Correction

After clustering the clusters and all the words were presented to the students and they were given the change to correct words that, according to them, did not belong to the correct cluster. In total seventeen corrections were given. In tabel 1 the corrections are visible. For each word it is indicated from what cluster it can and by whom, anonymized, it has been correct to which cluster the word originally belongs.

Value	Session	ClusterID	Corrected
agenda	a8c	1	4
bijslapen	ue2	3	2
fotograferen	a5b	2	3
gamen	da7	2	7
joggen	e65	2	6
kijken	e65	5	7
kijken	a8c	5	7
krachttraining	e65	3	6
ontspannen	q0f	6	2
ouders	n53	4	3
plannen	a8c	1	4
rennen	e65	2	6
slapen	e65	2	6
slapen	a5b	2	3
uitslapen	da7	2	3
voetbal	m1d	8	6
whiskey	p45	3	2

Tabel 1 Applied corrections

External analysis

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,804	0,875	0,673	0,696	0,684

Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4
agenda	bier	bierdrinken	boodschappen
plannen	borrelen	bijslapen	ouders
	drinken	familiebezoek	studeren
	eten	feest	studie
	fotograferen	krachttraining	
	gamen	stedentrip	
	gezellig	visite	
	internetten	whiskey	
	joggen		
	koffiedrinken		
	koken		
	lezen		
	relaxen		
	rennen		
	slapen		
	uitgaan		
	uitrusten		
	uitslapen		
	wandelen		
	wassen		
	winkelen		
	zuipen		

Cluster #5	Cluster #6	Cluster #7	Cluster 8
buiten	coachen	films	sport
doen	hardlopen	series	sporten
kijken	ontspannen	spelletjes	tennis
reizen	tennissen	tv	voetbal
stappen	voetballen		
varen			
werken			

Classes

Below all classes are presented, the classes are the result of the corrections given by the student and used as existing knowledge. After corrections one class remained empty, and was dropped.

Class #1	Class #2	Class #3	Class #4
agenda	bier	fotograferen	gamen
plannen	borrelen	bierdrinken	films
boodschappen	drinken	familiebezoek	series
studeren	eten	feest	spelletjes
studie	gezellig	stedentrip	tv
	internetten	visite	
	koffiedrinken	ouders	
	koken		
	lezen		
	relaxen		
	slapen		
	uitgaan		
	uitrusten		
	uitslapen		
	wandelen		
	wassen		
	winkelen		
	zuipen		
	bijslapen		
	whiskey		
	ontspannen		

Class #5	Class #6	Class #7
joggen	buiten	sport
rennen	doen	sporten
krachttraining	kijken	tennis
coachen	reizen	voetbal
hardlopen	stappen	
tennissen	varen	
voetballen	werken	

Appendix C – Second human informants experiment

Group 1

General information

- Vier studenten
- 50 woorden
- 39 unieke woorden
- 2 woorden niet herkend

Recognized words

bellen, bier, bijkomen, chatten, chillen, computeren, drinken, feesten, fietsen, fitness, gamen, kameraden, kater, keet, kijken, knuffelen, lachen, niksen, pils, plukken, praten, reizen, relaxen, rustig, slapen, socialiseren, sporten, televisie, tulpen, uitgaan, uitrusten, vervelen, voetbal, voetballen, volleyballen, vrienden, werken

Not recognized words

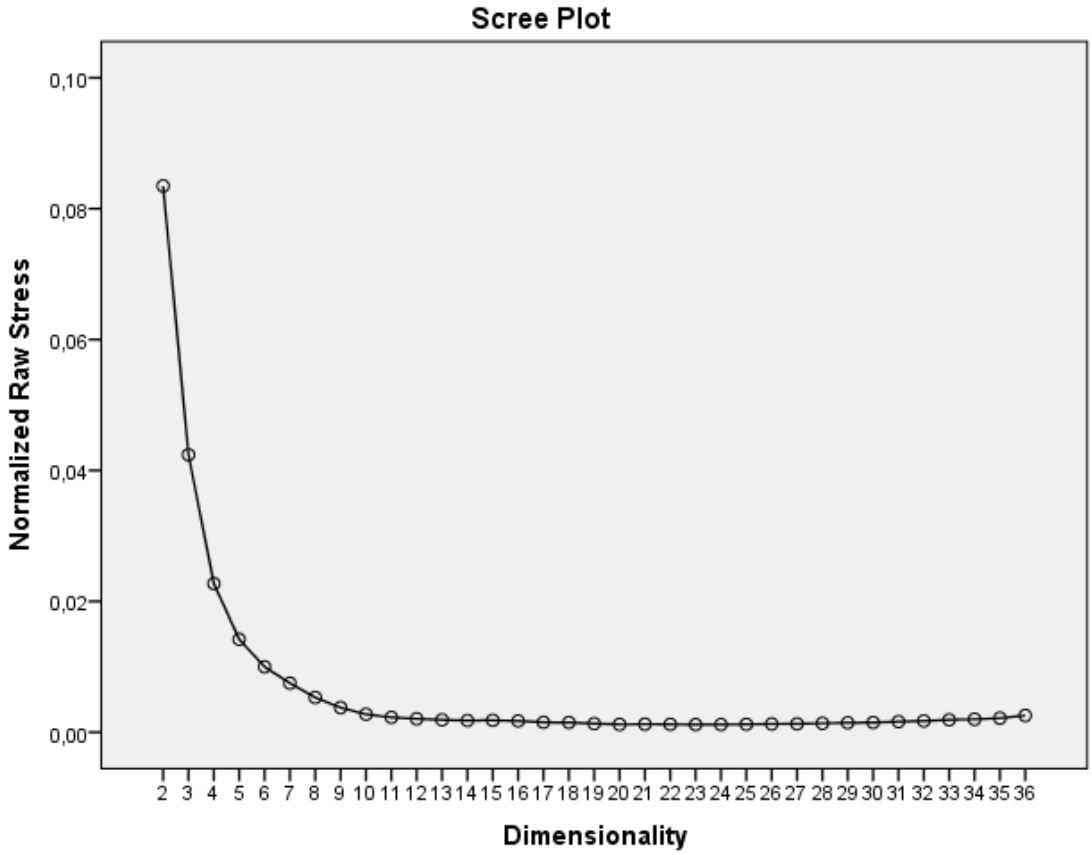
chaten, smsen

Classes

Class #1	Class #2	Class #3
Bier	Bellen	Bijkomen
Drinken	Chatten	Chillen
Feesten	Knuffelen	Computeren
Kameraden	Praten	Gamen
Kater		Kijken
Keet		Niksen
Lachen		Relaxen
Pils		Rustig
Socialiseren		Slapen
Uitgaan		Televisie
Vrienden		Uitrusten
		Vervelen

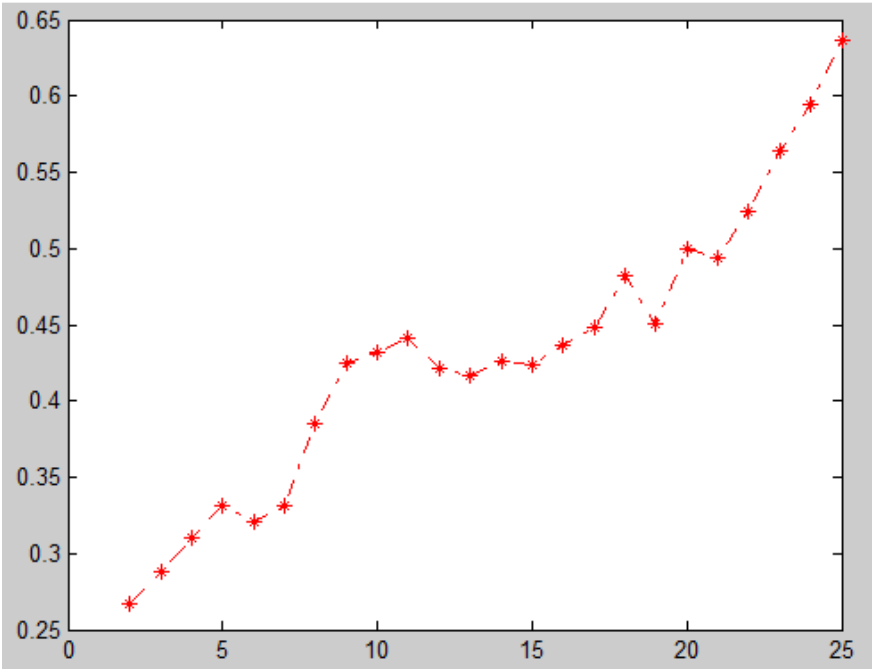
Class #4	Class #5	Class #6
Fietsen	Plukken	Reizen
Fitness	Tulpen	
Sporten	Werken	
Voetbal		
Voetballen		
Volleyballen		

Stress plot



Elbow method, stress is .01 at six dimensions

Silhouette plot



Clusters

Cluster #1	Cluster #2	Cluster #3
bellen	chatten	**bier**
computeren	**chillen**	kater
kijken	drinken	**keet**
lachen	gamen	**pils**
praten	knuffelen	tulpen
rustig	**niksen**	
slapen	plukken	
uitrusten	**relaxen**	
vervelen	socialiseren	
voetballen		
volleyballen		

Cluster #4	Cluster #5	Cluster #6
fitness	**feesten**	bijkomen
sporten	**kameraden**	**fietsen**
televisie	**vrienden**	**reizen**
voetbal		**uitgaan**
		werken

Measurements

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,378	0,713	0,308	0,255	0,279

Group 2

General information

- Vier studenten
- 40 woorden
- 27 unieke woorden
- 4 woorden niet herkend

Recognized words

consumeren, drinken, eten, feest, feesten, film, filosoferen, fotograferen, hardlopen, joggen, kerkbezoek, koffiedrinken, lezen, musical, natregenen, reizen, rennen, serie, shoppen, slapen, sporten, treinen, treinreizen, uitgaan, uitrusten, visite, zuipen

Not recognized words

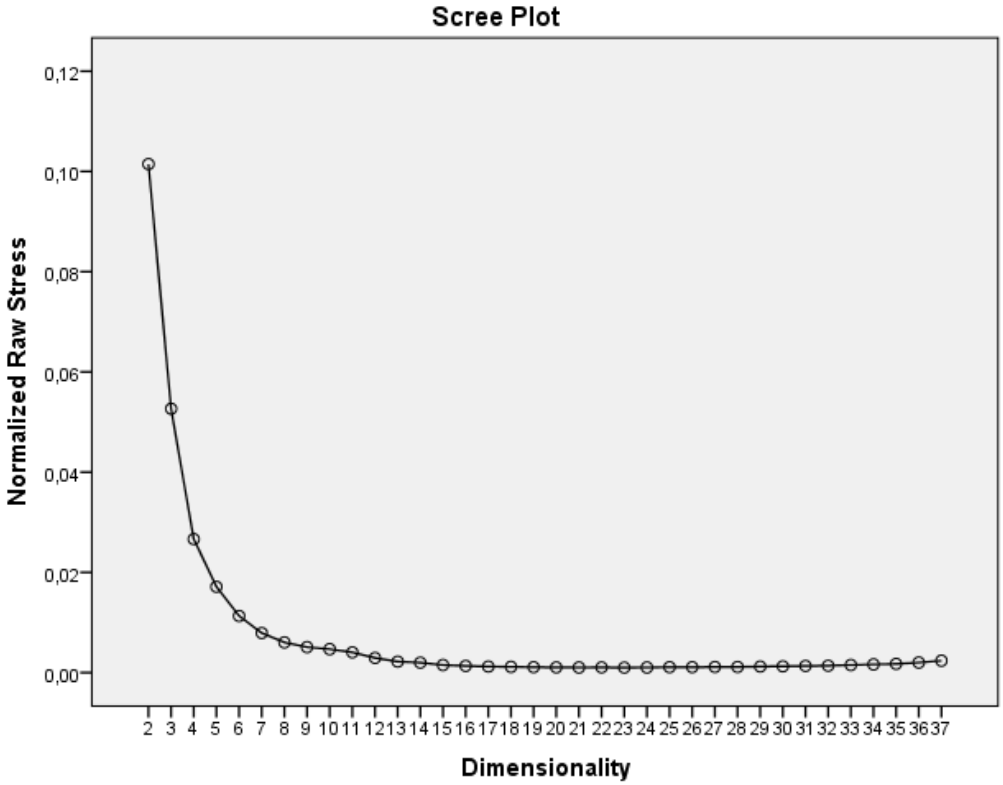
chickiescoren, eurovisionsongfestivallen, skutsjesilen, stroekendoeken, batavieren, polsstokverspringen

Classes

Class #1	Class #2	Class #3
Tackelen	Voetbal	Visite
Jagen	Pingpong	Feest
Schaken	Sporten	Zuipen
Drummen	Hardlopen	Feesten
	Joggen	Kerkbezoek
	Rennen	

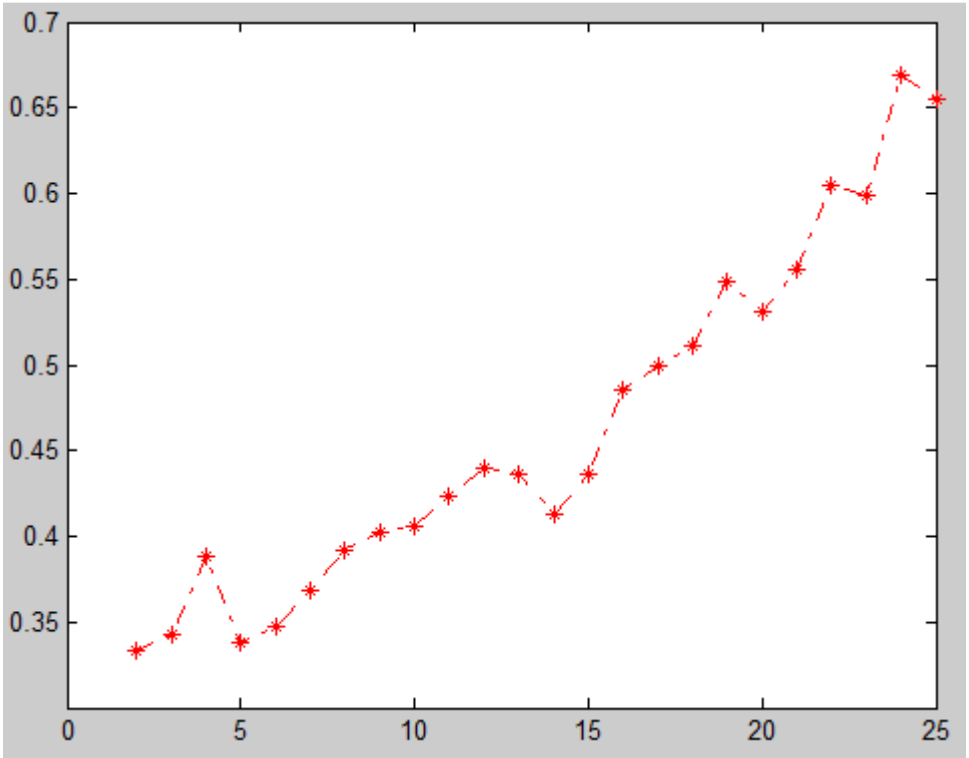
Class #4	Class #5	Class #6
Zuipen	Lezen	Pesten
Eten	Fotograferen	Poelen
Drinken	Filosoferen	Ruimen
Koffiedrinken	Uitrusten	Bergen
Consumeren	Treinreizen	Waken
	Treinen	Opvoeden
	Reizen	Baard
	Uitgaan	
	Natregenen	
	Shoppen	
	Film	
	Lied	
	Slapen	

Stress plot



Elbow method, stress is 0,005 at six dimensions

Silhouette plot



Clusters

Cluster #1	Cluster #2	Cluster #3
lezen	**schaken**	**drummen**
filosoferen	**voetbal**	**feest**
uitgaan	**pingpong**	film
natregenen	sporten	**lied**
waken		baard
opvoeden		

Cluster #4	Cluster #5	Cluster #6
hardlopen	**tackelen**	feesten
joggen	**jagen**	**treinreizen**
rennen	kerkbezoek	treinen
visite	consumeren	**reizen**
zuipen	**ruimen**	poelen
eten		**bergen**
drinken		
koffiedrinken		
fotograferen		
uitrusten		
shoppen		
pesten		

Measurements

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,282	0,724	0,213	0,210	0,211

Group 3

General information

- Vier studenten
- 47 woorden
- 20 unieke woorden
- 0 woorden niet herkend

Recognized words

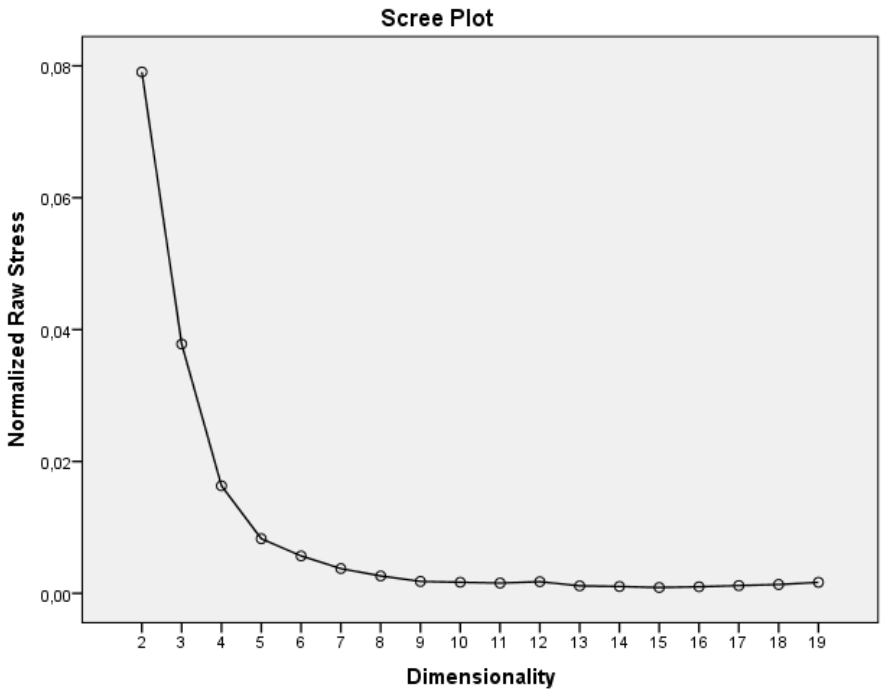
boodschappen, douchen, drinken, eten, familiebezoek, kijken, koken, lezen, ouders, reizen, schoonmaken, slapen, sporten, studeren, studie, treinreizen, tv, uitrusten, verjaardagen, werken

Not recognized words

Classes

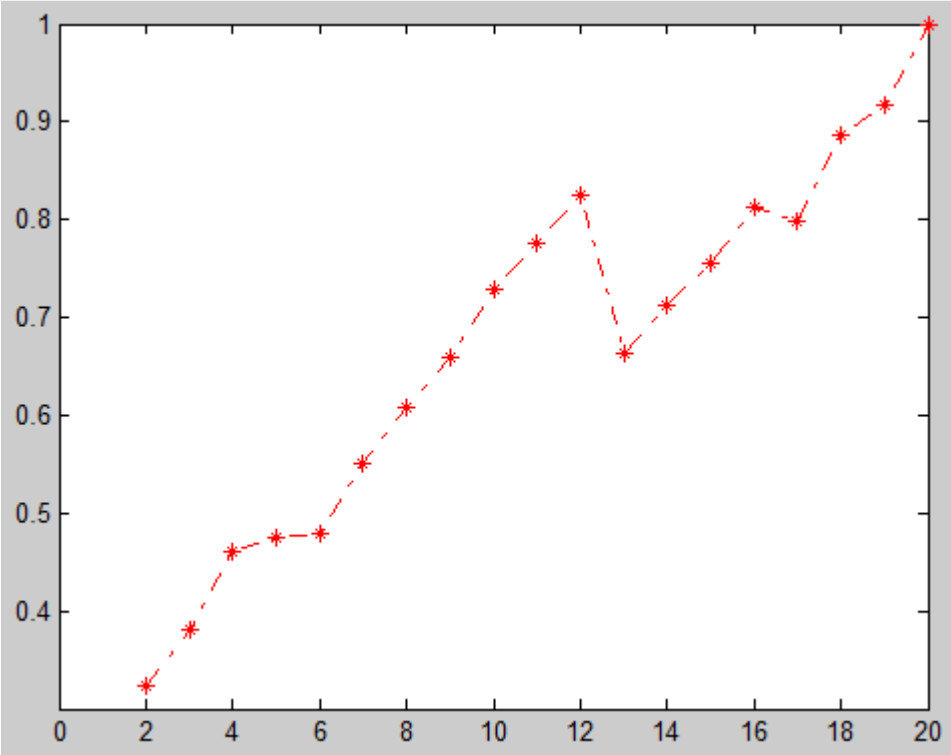
Class #1	Class #2	Class #3	Class #4
Slapen	Koken	Familiebezoek	Werken
Uitrusten	Eten	Ouders	Sporten
TV	Drinken	Treinreizen	Studeren
Kijken	Boodschappen	Reizen	Studie
Lezen		Verjaardagen	Schoonmaken
Douchen			

Stress plot



Elbow method, stress is .008 at five dimensions

Silhouette plot



Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4
familiebezoek	**sporten**	boodschappen	ouders
reizen	**tv**	douchen	**studeren**
treinreizen		**drinken**	**studie**
		eten	**werken**
		kijken	
		koken	
		lezen	
		schoonmaken	
		slapen	
		uitrusten	
		verjaardagen	

Measurements

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,350	0,674	0,338	0,537	0,415

Appendix D – Comparing personal clustering results

Orginal

Classes

Class #1	Class #2	Class #3	Class #4	Class #5	Class #6
Bier	Bellen	Bijkomen	Fietsen	Plukken	Reizen
Drinken	Chatten	Chillen	Fitness	Tulpen	
Feesten	Knuffelen	Computeren	Sporten	Werken	
Kameraden	Praten	Gamen	Voetbal		
Kater		Kijken	Voetballen		
Keet		Niksen	Volleyballen		
Lachen		Relaxen			
Pils		Rustig			
Socialiseren		Slapen			
Uitgaan		Televisie			
Vrienden		Uitrusten			
		Vervelen			

Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4	Cluster #5	Cluster #6
Bier	Bellen	Chatten	Fitness	Bijkomen	Feesten
Kater	Computeren	Chillen	Sporten	Fietsen	Kameraden
Keet	Kijken	Drinken	Televisie	Reizen	Vrienden
Pils	Lachen	Gamen	Voetbal	Uitgaan	
Tulpen	Praten	Knuffelen		Werken	
	Rustig	Niksen			
	Slapen	Plukken			
	Uitrusten	Relaxen			
	Vervelen	Socialiseren			
	Voetballen				
	Volleyballen				

Result

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,378	0,713	0,308	0,255	0,279

Person 1

Classes

Class #1	Class #2	Class #3	Class #4	Class #5	Class #6	Class #7
bellen	computeren	reizen	feesten	chillen	fitness	plukken
chatten	televisie	werken	drinken	knuffelen	sporten	tulpen
lachen	gamen		bier	rustig	voetbal	
praten	kijken		kater	slapen	voetballen	
socialiseren	niksen		keet	uitrusten	volleyballen	
vrienden	relaxen		pils	bijkomen	fietsen	
smsen	vervelen		uitgaan			
			kameraden			

Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4	Cluster #5	Cluster #6	Cluster #7
bellen	chillen	bijkomen	bier	kijken	fitness	feesten
chatten	gamen	reizen	kater	lachen	sporten	kameraden
computeren	knuffelen	werken	keet	praten	televisie	vrienden
drinken	niksen		pils	rustig	voetbal	
fietsen	plukken		tulpen	slapen		
uitgaan	relaxen			uitrusten		
	socialiseren			vervelen		
				voetballen		
				volleyballen		

Result

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,459	0,784	0,245	0,240	0,242

Persoon 2

Classes

Class #1	Class #2	Class #3	Class #4	Class #5	Class #6
bier	bijkomen	bellen	chatten	kijken	sporten
pils	chillen	praten	computeren	televisie	voetbal
drinken	relaxen		gamen		voetballen
kater	uitrusten				volleyballen
					fitness
					fietsen

Class #7	Class #8	Class #9	Class #10	Class #11
kameraden	feesten	tulpen	knuffelen	niksen
vrienden	uitgaan	plukken	rustig	vervelen
lachen			slapen	werken
socialiseren				
keet				
reizen				

Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4	Cluster #5	Cluster #6
bier	chillen	bellen	bijkomen	kijken	lachen
keet	gamen	chatten		praten	rustig
pils	knuffelen	computeren		reizen	slapen
tulpen	niksen	drinken		werken	uitrusten
	plukken	fietsen			vervelen
	relaxen	uitgaan			voetballen
					volleyballen

Cluster #7	Cluster #8	Cluster #9	Cluster #10	Cluster #11
kameraden	feesten	kater	sporten	televisie
vrienden	fitness		voetbal	
	socialiseren			

Result

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,297	0,836	0,103	0,127	0,114

Persoon 3

Classes

Class #1	Class #2	Class #3	Class #4	Class #5
bier	reizen	sporten	computeren	tulpen
drinken	bijkomen	fietsen	chatten	plukken
feesten	kijken	fitness	chillen	
kater	lachen	voetbal	gamen	
keet	relaxen	voetballen	televisie	
pils	socialiseren	volleyballen		
uitgaan				

Class #6	Class #7	Class #8	Class #9
slapen	bellen	knuffelen	werken
uitrusten	kameraden	rustig	vervelen
niksen	praten		
	vrienden		

Clusters

Cluster #1	Cluster #2	Cluster #3	Cluster #4	Cluster #5
bier	chillen	fitness	bellen	bijkomen
keet	gamen	sporten	chatten	
pils	knuffelen	televisie	computeren	
tulpen	niksen	voetbal	drinken	
	plukken		fietsen	
	relaxen		uitgaan	
	socialiseren			

Cluster #6	Cluster #7	Cluster #8	Cluster #9
lachen	feesten	kater	kijken
rustig	kameraden		praten
slapen	vrienden		reizen
uitrusten			werken
vervelen			
voetballen			
volleyballen			

Result

Name	Purity	RandIndex	Precision	Recall	F1-Score
Word2Vec	0,405	0,815	0,179	0,192	0,185