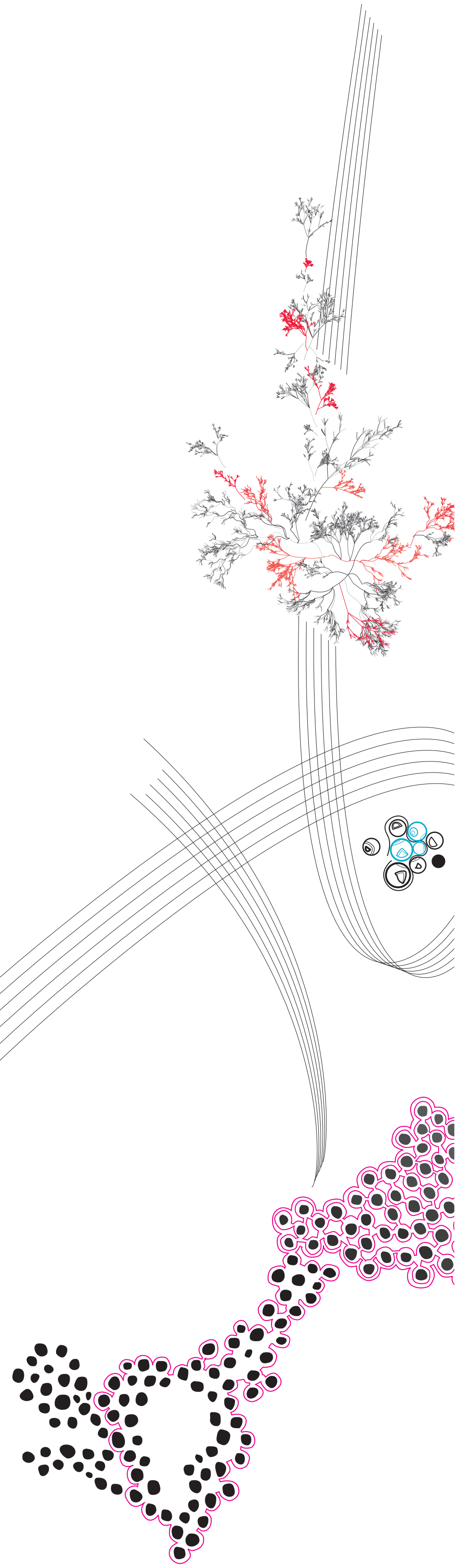

MSc Thesis Computer Science

Building a Sense Inventory for Dutch Healthcare Abbreviations

Guido A.M. van der Heijden

Supervisors
Shenghui Wang (University of Twente)
Dolf Trieschnigg (Nedap)
Doina Bucur (University of Twente)

September 14, 2022

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

My graduation project has been filled with challenges. The limited availability of expert annotators and the uncertain properties of the abbreviations created a lot of challenges, which resulted in ups and downs in my motivation. The thesis presented a causality problem: I did not know whether Dutch healthcare abbreviations have multiple definitions, which is a necessary property for this thesis to be useful, but a dataset of abbreviations needs to be made to know whether this property of abbreviations holds. At times, I felt like there weren't any ambiguous abbreviations and that the premise of my thesis was falling apart.

Fortunately, I was motivated to keep on going by my supervisors, colleagues at Nedap, family, and partner. I am especially grateful for the push from my supervisors to analyze my results more extensively, which allowed for many interesting observations. Furthermore, I greatly appreciate the annotation effort by the annotators: Tristan de Boer, Max Driessen, Joost Holslag, Marloes Nijman, Thomas Markus, Linda Meijer, Luuk Meijerink, Guido van Schie, Dolf Trieschnigg, Bastienne van der Zanden. Annotating abbreviations was a relatively mundane task, as I personally experienced through looking at them for hundreds of hours. Lastly, I like to thank Robin de Groot for providing insightful feedback on a thesis draft.

Abstract

Healthcare abbreviations pose problems to people reading healthcare reports and to text mining, due to being unknown or ambiguous. Word sense disambiguation (WSD) has been used to tackle the ambiguity of abbreviations, but WSD is bound by the exhaustiveness of abbreviation sense inventories. Unsupervised WSD, more often referred to as word sense induction (WSI), has been proposed to overcome the inhibiting dependency on sense inventories.

A sense inventory can be constructed by annotating randomly sampled abbreviation occurrences, but this is a cumbersome approach. This thesis explores whether WSI can be used to reduce the annotation cost for finding abbreviation senses, while maintaining high sense coverage. In this thesis, WSI entails clustering vectorized abbreviation occurrences, with the aim of grouping together the occurrences of the same sense. Each cluster centroid is then annotated with a sense, which related work has shown to reduce the number of annotations needed to retrieve an abbreviation's senses. WSI is conducted using three different vectorization methods to represent an abbreviation occurrence's context: pointwise-mutual-information-weighted bag-of-words, maximum surrounding-based embedding, and substitution lemmas using a RoBERTa model pretrained on Dutch hospital notes (MedRoBERTa.nl). The vectorized abbreviation occurrences are clustered using two clustering methods: Tight Clustering for Rare Senses (TCRS) [45], and k-means. Clustering substitution lemmas with TCRS results in the greatest reduction in annotation cost (34.5%) with respect to a random annotation baseline, while obtaining high sense coverage (87.0%).

Aside from clustering, abbreviation occurrences and occurrences of candidate senses are compared through two sentence similarity measures: Word Mover's Distance, and cosine similarity of summed word embeddings. I use these measures to rank the candidate senses from most likely to least likely being the sense of a group of abbreviation occurrences. The rank of the exact sense is quite bad, but the highest-ranking candidate senses are often inflections are synonyms of the exact sense. This indicates that semantic similarity ranking can be used for building a sense inventory, but adaptations and additions to my method are necessary.

Keywords: healthcare abbreviations, word sense induction, word sense disambiguation, clustering, word embeddings, MedRoBERTa.nl, semantic similarity, word mover's distance.

Contents

1 Introduction

Healthcare information, such as the content of an electronic health record, is often recorded as unstructured free text. This free text can be difficult to process for natural language processing (NLP; an index of abbreviations is given on p. 55) systems and human readers due to spelling errors, incorrect punctuation, domain-specific vocabulary, and abbreviations [17, 13].

An abbreviation can be considered as a homonym, since it is spelled and pronounced the same for each meaning, but it can have multiple unrelated meanings [39]. These meanings are referred to as *senses* in NLP literature. Senses can be defined in various ways, such as glosses or complete definitions. In this thesis, the long forms (sometimes referred to in literature as written-out or full form) of an abbreviation are considered its senses, e.g. the long forms of "*ab*" can be "*antibiotica* (antibiotics)" and "*activiteitenbegeleider* (activity counselor)". In contrast to homonyms, a polyseme has multiple, related yet differentiable, senses. For each sense, the word is spelled the same, but it can be pronounced differently. For instance, a "*dish*" can be both a "*plate from which you can eat*", a "*meal that you can eat*" or even an "*attractive woman*".[1] These senses can be differentiated, but are related in their linguistic history, namely in their etymology.

Abbreviations cause problems due to 1) occasionally having multiple senses (i.e. being ambiguous), and 2) not being covered by a sense inventory (i.e. the sense being unknown) [42, 25, 45, 44]. The former problem, abbreviation ambiguity, showed to cause 8.4% of medication errors in Australian hospital notes [10]. The misinterpretation of these abbreviations mainly led to wrong medication dosages, of which 29.6% were considered high risk for causing significant harm. The latter problem, abbreviation senses not being known, inhibits communication among healthcare professionals. It has been reported that domain-specific abbreviations are known to domain experts for pediatric notes [22], but a more extensive audit showed that abbreviation understanding decreased when a healthcare professional was less associated with the abbreviation domain [33]. A quiz showed that four types of professionals outside pediatrics recognized 41.25% of abbreviations used in pediatric reports (handover sheets and medical notes) on average, and this was 67% on average for four types of pediatric professionals. Here, the pediatric consultant recognized most, and had 90% correct. This was remarkable, as the pediatric consultant handled pediatric reports on a daily basis, and still was not able to recognize 10% of the abbreviations in their field.

A comprehensive sense inventory for Dutch healthcare abbreviations allows for mitigating the problems caused by abbreviations to NLP and human readers [25]. Such a sense inventory is unfortunately not available, but can be constructed from a corpus containing occurrences of healthcare abbreviations. A cumbersome approach would be to build a sense inventory by annotating many randomly sampled abbreviation occurrences with their sense [25, 45]. Each uniquely identified abbreviation sense is then incorporated in the sense inventory. The downside of this approach is that this requires a substantial manual workload by experts [28]. For example, Moon et al. [25] had two experts annotate 500 randomly selected occurrences per abbreviation for 440 abbreviations. The abbreviations in their sense inventory had 2.16 senses on average, where 62.7% of abbreviations only had a single sense and 21.7% of abbreviations had a majority sense with a frequency

[1] Etymology of "*dish*"

over 95%.

Xu et al. [45] formulated an approach to reduce the number of annotations required for extracting abbreviation senses from a corpus. They clustered the occurrences of English clinical abbreviations based on their contexts. The idea was that, since abbreviations are considered homonyms rather than polysemes, each cluster of contextually similar abbreviation occurrences should only contain occurrences of the same sense. Therefore, a single occurrence per cluster could be annotated to retrieve the various senses of an abbreviation, instead of annotating a larger number of random occurrences. The clustering approach for retrieving abbreviation senses by Xu et al. is the main inspiration for this thesis. This thesis mainly differs from the work done by Xu et al. in its focus on Dutch healthcare abbreviations, rather than English clinical abbreviations, and by exploring state-of-the-art methods for clustering the abbreviation occurrences.

Word sense disambiguation is "*the computational identification of meaning for words in context*" [28, p. 2]. The clustering approach by Xu et al. [45] is referred to in literature as word sense induction (WSI), which is the subset of word sense disambiguation (WSD) that does not rely on annotated datapoints, i.e. WSI is unsupervised WSD. WSI and WSD will be addressed in further detail in Section 2. This thesis explores to what extent WSI can aid in reducing *annotation cost*, which is the number of annotations used for extracting Dutch healthcare abbreviation senses. This is expressed as *annotation cost reduction*, which is the reduction in the number of annotations needed when using a WSI approach compared to random annotation for retrieving the same number of senses. Similarly, the improvement in *sense coverage* will be measured as *sense coverage gain*. Sense coverage is the fraction of senses a WSI approach can obtain compared to a gold standard. Sense coverage gain is the gain in sense coverage when using a WSI approach compared to random annotation while annotating the same number of occurrences. These metrics are more elaborately described in Section 5.1.1.

Some studies showed that the semantic similarity between an abbreviation occurrence and occurrences of its candidate senses can be used to disambiguate the abbreviation [19, 6]. Candidate senses are n-grams that might be the sense of an abbreviation. The premise here is that the context of an abbreviation is more similar to the context in which its long form occurs, than to the contexts in which the other candidate senses of the abbreviation occur. This thesis will explore to what extent semantic similarity can be used to rank candidate senses of an abbreviation, and whether that shows improvement compared to randomly ranking these candidate senses.

1.1 Research Question

This thesis sets out to build a sense inventory for Dutch healthcare abbreviations. Therefore, I pose the following research question and sub-questions:

RQ How can a sense inventory be build for abbreviations from Dutch healthcare reports?

sub-RQ1 To what extent can WSI reduce annotation cost for extracting senses compared to annotating random abbreviation occurrences?

sub-RQ2 How much gain in sense coverage can be achieved by annotating WSI clusters compared to annotating random abbreviation occurrences?

sub-RQ3 To what extent can semantic similarity measures accurately rank candidate senses of an abbreviation?

The main research question is tackled using WSI and semantic similarity approaches. The first two sub-questions regard the performance of the WSI approach, whereas the third sub-question regards the viability of semantic similarity measures in ranking candidate senses. Firstly, annotating randomly selected abbreviation occurrences is inefficient, and exploring the extent to which WSI can reduce the annotation cost is relevant with respect to the main research question. Secondly, it is possible that not every sense of an abbreviation is captured by a WSI cluster, thus sense coverage achieved through WSI is relevant with respect to the main research question. Thirdly and lastly, the capabilities of semantic similarity in ranking candidate senses to abbreviation occurrences is explored, since semantic similarity might have the potential to aid in retrieving an abbreviation's senses without the need for any annotation.

1.2 Remainder of this Thesis

This thesis continues by presenting Background literature and Related Work, and the role it has in this thesis (Section 2). The Method (Section 3) consists of two parts: 1) vectorization and clustering methods for WSI, 2) semantic similarity methods for ranking candidate senses. Vectorization entails the transformation of the context of an abbreviation occurrence into numerical vectors, e.g. by using a weighted bag-of-words for the context words, or by using domain-specific word embeddings. These vectorized abbreviation occurrences are then clustered with the aim of clustering together occurrences of the same sense. The semantic similarity methods are used to calculate the similarity between two sentences based on the word embeddings of each sentence.

I created a Dataset (Section 4) containing the senses of 17 Dutch healthcare abbreviations with the help of domain experts. The experts annotated a sample of 60 occurrences of each abbreviation in deidentified healthcare reports from Nedap Healthcare. 5 of the abbreviations originate from healthcare reports of the elderly care sector, 6 from the mental healthcare sector, and 6 from the disabled care sector. The frequency of each sense relative to its abbreviation is also included. The Experiments (Section 5) go into the technical details of preprocessing the healthcare reports, and how the methods from Section 3 are applied to the dataset. The Evaluation (Section 6) shows that not all WSI methods are successful, and candidate sense ranking does not yet appear viable for retrieving senses of Dutch healthcare abbreviations. The best performing WSI method entailed using a domain-specific language model, called MedRoBERTa.nl, to predict substitutes of abbreviations and clustering those substitutes. An extensive error analysis addresses the pitfalls of the best WSI method, and the inconsistent performance of candidate sense ranking. The Discussion (Section 7) addresses practical implications of this thesis, limitations, and suggestions for future work. The Conclusion (Section 8) summarizes the main findings and relates those back to the research question and sub-questions.

2 Background and Related Work

This section contains a mix of background literature and related work to aid the reader in better understanding the topic of this thesis, and to provide a foundation for the Method in Section 3. Firstly, Sections 2.1 and 2.2 provide theoretical background on language models & embeddings and word mover's distance respectively. If these topics are familiar to the reader, these sections can be skipped. However, it is recommended to read Section 2.1.1, since it contains information on the Dutch medical language transformer model (MedRoBERTa.nl) that is used in this thesis. Secondly, more background is given on WSI in Section 2.3, and its subsections provide a background on the various types of WSI. Sections 2.3.1 and 2.3.4 discuss related work of two types of WSI that are used in this thesis. Thirdly and lastly, Section 2.4 contains related work that used semantic similarity to normalize abbreviations.

2.1 Language Models & Embeddings

Embeddings are a numerical representation of words, parts of words, word groups, sentences, or even paragraphs. These embeddings aim to capture the semantic representation of a piece of text, such that the embedding can be used as a starting point for a wide variety of NLP tasks. Embeddings substitute the need for extensive feature engineering, such as performed in feature-based WSI (Section 2.3.1). The skip-gram model for generating word embeddings has the objective to maximize the average log probability that a word occurs in the context of other words [24]. This objective is maximized by a neural network, where the input is a one-hot-encoding of the word at the center of a sequence of words, and the output consists of one-hot-encodings of the surrounding words. The hidden layer of this network has a dimensionality of $V \times D$, where V is the size of the vocabulary and D the dimensionality of each word embedding. In other words, the skip-gram model is used to create D-dimensional numerical representations of words based on the context that those words occur in.

BERT, a deep bidirectional transformer neural network, greatly improved results of NLP tasks by creating embeddings through a different approach [8]. Contrary to skip-gram, BERT is better at taking into account the direction of context words, and it uses WordPiece tokenization rather than considering each word a token. WordPiece tokenization entails that the vocabulary includes tokens for parts of a word, and for whole words. This results in better embeddings of rare words and words that are not included in the vocabulary. One of the tasks used to pre-train BERT is the fill-mask task. This task entails that a sentence is used as input to BERT where a token is replaced with `<mask>`, and the correct output is the masked token. This task requires no training data, yet allows BERT to capture the semantics of a sentence and the tokens in it.

Liu et al. [20] introduced RoBERTa (Robustly optimized BERT approach), which entails an improved pre-training procedure of BERT that results in better performance when used for various NLP tasks. The design decisions for pre-training included: training the model longer; using larger batch sizes; removing the next sentence prediction objective; training on longer text segments; and dynamically changing the pattern for the fill-mask task during training.

2.1.1 MedRoBERTa.nl

Verkijk and Vossen [38] set out to create the first transformer-based language model for Dutch medical language. They argued for the need of a domain-specific language model, since such models had shown to obtain a better semantic understanding of text for tasks within the domain they were trained on. Verkijk and Vossen pointed out that hospital notes displayed distinctive characteristics from other Dutch texts. The notes were characterized by simplified sentences, a lack of attention to grammar, mixing general and specialized language, and repurposing words to create new lingo. Due to those characteristics, Dutch hospital notes could benefit from a domain-specific language model, rather than a general Dutch language model.

Verkijk and Vossen stipulated a method in which they compared general Dutch (Ro)-BERT models, a RoBERT model pre-trained from scratch on Dutch hospital notes (MedRoBERTa.nl), and a RoBERT model extended by training it on hospital notes. They formulated three evaluation tasks: an intrinsic in-domain sentence similarity task, an extrinsic in-domain sentence classification task, and an extrinsic out-domain NER task. Firstly, the sentence similarity task entailed identifying the odd-one-out sentence from a set of three sentences. MedRoBERTa.nl outperformed the other transformers in terms of accuracy on the sentence similarity task (0.65 versus 0.52-0.58 respectively). Secondly, the sentence classification task entailed predicting a sentence's class among 4 domain-specific classes and an 'other'-class. All transformers performed with insignificant difference from one another, but MedRoBERTa.nl obtained a significantly higher F1-score for one of the classes. Lastly, the NER task entailed tagging entities in a Dutch newspaper texts. Here, the general transformers obtained much better F1-scores than MedRoBERTa.nl (0.84-0.91 and 0.66 respectively). This was to be expected, since MedRoBERTa.nl was designed for medical NLP tasks, which made it less applicable to general Dutch NLP tasks.

The results of these evaluation tasks show that MedRoBERTa.nl is the best model to use on Dutch medical text. Since the healthcare and medical domain are closely related, and MedRoBERTa.nl was also trained on patient reports, MedRoBERTa.nl is a suitable model for embedding-based WSI for Dutch healthcare abbreviations. Related work on embedding-based WSI is discussed in Section 2.3.4.

2.2 Word Mover's Distance

Kusner et al. [18] introduced a novel metric for measuring the dissimilarity between two text documents. They denoted that word embeddings for similar words are often close to each other within an embedding space, while dissimilar words are more distant. Based on that observation, they argued that the dissimilarity of two documents could be measured as the minimum cumulative distance of non-stopwords from one document to another document. They coined this metric as Word Mover's Distance (WMD), since it could be solved similar to the Earth Mover's Distance, a transportation optimization problem. WMD is shown visually in Figure 1 for one document containing an abbreviation, and another document containing the sense of that abbreviation. When there are more words in one document than the other, the mapping of words is represented as flow. In that case, each word from one document containing N non-stopword words requires $1/N$ outgoing flow, and each word of another document containing M non-stopword words requires

$1/M$ incoming flow.

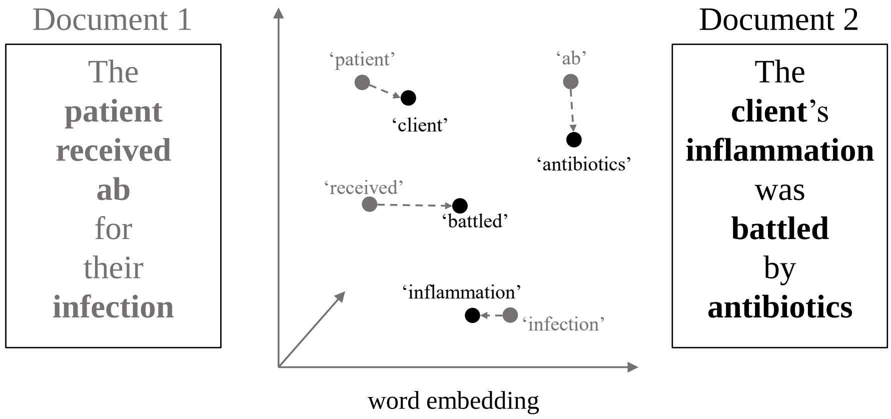

FIGURE 1: Visualization of an optimal solution for WMD given two documents. The middle shows a plot containing the non-stopwords of two documents in a word embeddings space. An optimal mapping is given via arrows, which results in the minimum cumulative distance between the words in document 1 with respect to document 2.

WMD is computationally relatively expensive, namely time complexity $O(p^3 \log p)$, where p is the number of unique words in each document. Therefore, two faster variants of WMD were formulated: Word Centroid Distance (WCD), and relaxed WMD [18]. WCD entails representing the distance between two documents as the euclidean distance between the averaged word embeddings of each document. WCD can be computed very fast ($O(p)$), but does not have tight bounds with respect to WMD. Relaxed WMD entailed relaxing the constraints for optimizing WMD. Rather than mapping the words in each document using flow, each word from one document could be mapped to any word of the other document. Relaxed WMD thus became the cumulative distance between each word in one document and the closest word in another document (see Equation 1). Relaxed WMD has time complexity $O(p^2)$.

$$\texttt{relaxedWMD}(D_1, D_2) = \sum_{w_i \in D_1} \min_{w_j \in D_2} \text{dist}(e(w_i), e(w_j)) \tag{1}$$

WMD and its faster variants were evaluated on document classification using the k-nearest-neighbour decision rule for 8 different corpora, and compared to 7 baselines including TF-IDF, LSI, BM25 Okapi, and LDA. WMD outperformed all other baselines for 6 corpora, and performed worse and on par with LSI for two corpora. WCD showed better performance than 6 baselines, while relaxed WMD almost performed equally to WMD.

2.3 Word Sense Induction

Abbreviations have not only been problematic within the healthcare sector. Abbreviation extraction and normalization has also been studied for biomedical papers. Abbreviation normalization in biomedical papers relied on the co-occurrence of the first appearance of an abbreviation with its long form [3, 47, 37, 46, 31]. Pattern-matching approaches looked for where an abbreviation occurred between parentheses to locate its sense. Instead of pattern-matching algorithms, studies on clinical abbreviation disambiguation employed supervised WSD [27, 40, 41, 34]. Supervised WSD is regarded as a text classification problem with a finite set of classes, similar to part-of-speech tagging or named-entity recognition [28, 39]. It is dependent on the comprehensiveness of a sense inventory and on sufficient labelled data. Unfortunately, those resources are not readily available for the disambiguation of Dutch healthcare abbreviations.

WSI aims to overcome the limitations of supervised WSD [28, 21]. WSI entails context clustering, and it is grounded in the assumption that the sense of a word should be derivable by the semantics of its context. By clustering together occurrences of a word with a (semantically) similar context, WSI discriminates between senses. In this thesis, the goal of using WSI is to derive a sense inventory for abbreviations, but WSI has also been used for disambiguation. Manandhar et al. [21] formulated a WSI task for the SemEval2010 conference that evaluated WSI. Unfortunately, problems arose due to it being difficult to map induced senses to human-defined senses. This caused evaluation metrics that automatically linked context clusters to appropriate senses to favor WSI methods that mapped all instances of a word to a single cluster, or WSI methods that mapped each instance to its own cluster.

Jurgens and Klapaftis [15] formulated another WSI task for SemEval2013 that took into account graded senses, where the instance of a word could have multiple senses at once, such as in double entendres. For example, in the phrase "*you look really hot!*", the writer/speaker hides a flirtatious insinuation (the receiver being attractive) behind an innocent remark (the receiver being overheated). Though abbreviations could be ambiguous due to a lack of context to disambiguate them, a writer of healthcare reports does not intend an abbreviation to be ambiguous. Therefore, it is not necessary to consider graded senses for healthcare abbreviation WSI. This means that WSI for abbreviations does not require fuzzy clustering, nor fuzzy evaluation metrics.

WSI often consists of two components: 1) a transformation of a word's context to input data, 2) and a clustering algorithm. The remainder of this subsection outlines four approaches found in literature on WSI, which are split up based on what input data is used: vectorized features based on a word's context (Section 2.3.1), a graph capturing the co-occurrence of context words (Section 2.3.2), topics found by supplying context words to a topic-model (Section 2.3.3), and vectorized context words through word embeddings (Section 2.3.4).

Before continuing with elaborating on WSI approaches, it is important to know how WSI can be used to derive abbreviation senses. Xu et al. [45] specified how sense clusters, which are the clusters obtained through WSI, are assigned a sense. In the best case scenario, when all occurrences of a sense are contained in a single cluster, a single occurrence of the cluster can be annotated to label said cluster with a sense. However, an imperfect clustering results in impure clusters, where some occurrences have one sense, and some have another sense. In this case, the occurrence that lies closest to the cluster's center, i.e. the centroid, would be best to annotate, since it should be most representative of the sense cluster. This method of sense assignment is depicted in Figure 2, and shows how WSI can aid in building a sense inventory for Dutch healthcare abbreviations.

All WSI approaches use the words in the context of a target word. Moon et al. [25] explored what window size for context words works best for WSD. A window size of 10-60 words on both sides of an abbreviation resulted in a support vector machine (SVM)

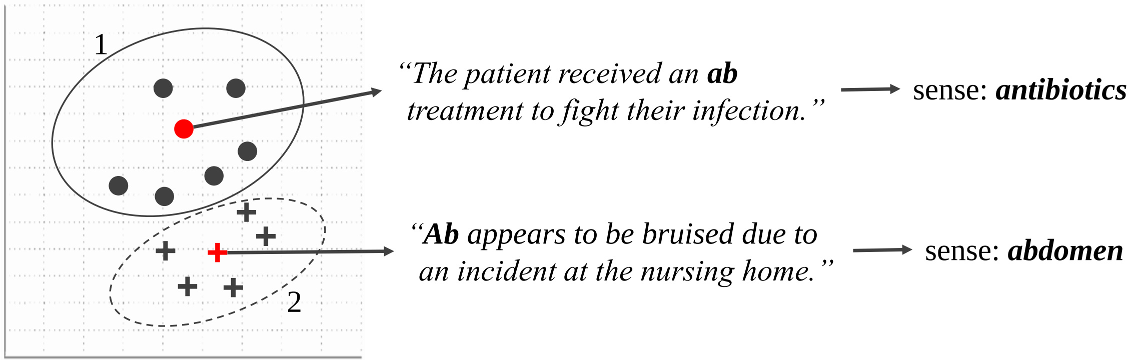
1
2
"The patient received an **ab** treatment to fight their infection."
sense: ***antibiotics***
"**Ab** appears to be bruised due to an incident at the nursing home."
sense: ***abdomen***

FIGURE 2: Visualization of how sense clusters can be assigned a sense for the abbreviation "*ab*". Each cluster consists of occurrences that are all the same sense. The centroid occurrence (red) of cluster 1 is annotated as "*antibiotics*", which means that cluster 1 is assigned the sense "*antibiotics*". Similarly, cluster 2 is assigned the sense "*abdomen*".

that was about 92% accurate in disambiguating abbreviations. Meanwhile, using only a window size of 5 reduced the accuracy to 90.8%, a window size of 3 to 89.5%, and using more than 60 words for the context window reduced the accuracy back to 90.0%. They also showed that it was important to use both sides of an abbreviation's context, since using only the right side reduced accuracy by 6 percentage points, and using only the left side by 2.5 percentage points.

2.3.1 Feature-based Approaches

Feature-based WSI and WSD represent the context of a target word as a bag-of-words [21, 30, 45]. Different weightings are used, such as point-wise mutual information (pmi) [45] (see Equation 2) or TF-IDF weighting [30]. Furthermore, some studies took into account the distance of a context word with respect to the abbreviation. Finley et al. [11] used a sigmoid function to decrease the weight of context words when they were more distant from an abbreviation. Similarly, Xu et al. [45] captured the positional information of context words by adding '*L*' or '*R*', together with a number, to the string of a context word, and included those strings into their bag-of-words vocabulary. For example, "*received*" would be represented with positional information as "*L2_received*" for the occurrence shown in Figure 2.

Xu et al. [45] is a particularly interesting case of feature-based WSI, since they evaluated their clustering models in the context of building a sense inventory. Their WSI method is used in this thesis with slight adaptations, because they showed that it leads to annotation cost reduction and sense coverage gain. Their method is therefore extensively discussed in the following paragraphs.

Xu et al. [45] applied feature-based WSI to extract English clinical abbreviation senses. The features used for clustering were the stemmed words within a window of 5 words from the abbreviation occurrence, both with and without positional information, and the section title of the admission note. The stemmed words were weighted using point-wise mutual information (see Equation 2). Here, w was a context word and a an abbreviation. The pmi was known to be biased towards infrequent words, so it was multiplied by a discounting factor (see Equation 4).

$$pmi(w,a) = \log_2 \frac{p(w,a)}{p(w) \cdot p(a)} = \log_2 \frac{\frac{c(w,a)}{N}}{\frac{c(w)}{N} \cdot \frac{c(a)}{N}} \tag{2}$$

$$= \log_2 \frac{N \cdot c(w,a)}{c(w) \cdot c(a)} \tag{3}$$

$$df(w,a) = \frac{c(w,a)}{c(w,a)+1} \cdot \frac{\min[c(w),c(a)]}{\min[c(w),c(a)]+1} \tag{4}$$

To cluster these features, Xu et al. [45] developed a clustering method for capturing rare senses (frequency $< 2\%$) of abbreviations, called Tight Clustering for Rare Senses (TCRS). Previously, they used the expectation maximization (EM) algorithm for sense clustering [43], but found through error analysis that EM clustering failed to recognize rare senses. The TCRS algorithm consisted of two phases: 1) finding tight clusters where all datapoints had a cosine similarity greater than a threshold θ_1, and 2) merging clusters using complete linkage based on a threshold θ_2. Complete linkage entailed that two clusters were merged when the minimum similarity between every point among the clusters was greater than a threshold θ_2. The clusters were merged iteratively, starting with the most similar clusters. This was done until either the number of clusters was less than a predefined number N, or the remaining clusters were not eligible for complete linkage due to their similarity being below θ_2. The thresholds (θ_1 and θ_2) were optimized using a set of 12 abbreviations, which resulted in $\theta_1 = 0.65$ and $\theta_2 = 0.1$.

A comparison was made between EM clustering and TCRS using a gold standard consisting of another 12 abbreviations (similar to the gold standard that is shown in Table 6). The gold standard was constructed from 200 randomly selected annotated abbreviation occurrences, which provided information on the relative frequency for each sense of an abbreviation. To evaluate the clustering methods, clusters were assigned a sense based on the gold standard, instead of assigning a sense to each cluster by annotating its centroid. This prevented the annotation effort from becoming too large. When the gold standard occurrences within a cluster were more than 60% of the same sense, this sense would be assigned to the cluster. For example, in Table 1, cluster-1 would be assigned the sense "*bowel movement*", cluster-2 would be assigned the sense "*bone marrow*", and cluster-3 would have no sense.

Sense/cluster	cluster-1	cluster-2	cluster-3
bowel movement	150	5	5
bone marrow	5	30	5

TABLE 1: A fictional example for clustering 200 occurrences of the abbreviation "*bm*", which has the possible senses "*bowel movement*" and "*bone marrow*".

EM clustering and TCRS were compared based on annotation cost and sense coverage. The annotation cost was defined relative to the gold standard of 200 annotations, which meant that a clustering algorithm that yielded 10 sense clusters would have an annotation cost of $200/10 = 0.05$. Sense coverage was also defined relative to the gold standard, which meant that the sense coverage was the number of senses retrieved by a clustering method over the number of senses in the gold standard. For example, the abbreviation "*ss*" had 7 senses in the gold standard, and TCRS managed to retrieve 6 senses. The sense coverage then became $6/7 = 0.86$.

On average, TCRS had an annotation cost of 0.069, and EM had an annotation cost of 0.050. This resulted in a sense coverage of 0.71 for TCRS and 0.56 for EM clustering averaged over 10 repetitions of clustering the 12 abbreviations. Furthermore, the clustering methods were compared to annotating a random sample of 10 occurrences (equivalent to an annotation cost of 0.050), which merely obtained 0.61 sense completeness.

It should be noted that the definition of annotation cost was actually flawed, but did not impede the results of Xu et al. [45], because each abbreviation had the same number of annotations in the gold standard. However, it is illogical to bind annotation cost to the gold standard, since changing the number of annotations in the gold standard should not affect the extent to which WSI can reduce the number of annotations to find an abbreviation's senses. For example, if only 50 occurrences of the abbreviation "*ss*" were included in the gold standard, the annotation cost for all methods would increase with over 100%. However, the absolute number of annotations required by each method would not change. It would have been better to measure the reduction in annotation cost of the clustering methods relative to the random annotation baseline, i.e. reduction in annotation cost through WSI. Therefore, this thesis emphasizes annotation cost reduction rather than absolute annotation cost.

As was the goal behind the design of TCRS, it managed to identify rare senses. From the 15 senses with a frequency of <2%, TCRS identified 7, while EM and random sampling were only able to identify 1. Meanwhile, all three of the methods were able to identify 26/26 abbreviations that had a frequency of >10%. To summarize, feature-based vectorization and TCRS achieved better sense coverage for a similar annotation cost as other methods, which makes it an interesting WSI method for retrieving abbreviation senses.

2.3.2 Graph-based Approaches

Graph-based WSI entails building a graph from a target word's occurrences, where the initial vertices are the words that co-occurred with the target word [32, 9, 5]. An edge is added for each word that co-occurs, and a vertex for a word is added if it does not occur in the graph yet. Sometimes, the edges are assigned weights to indicate a stronger relationship between words [9]. Then, this graph is split up into multiple sub-graphs by removing the edges, such that each sub-graph represents a sense. The similarities between the words in each sub-graph and the words in an abbreviation occurrence are used to disambiguate between the senses of a word, i.e. each abbreviation occurrence is linked to its most similar sub-graph for disambiguation.

Firstly, a significant limitation of graph-based WSI to building a sense inventory is that it becomes difficult to assign a sense to each sub-graph. For example, a sub-graph originating from occurrences of the abbreviation "*ab*" can be "{*disease, cure, infection, medication, prescription*}". This sub-graph can represent the sense "*antibiotics*", but it can also represent some brand of medication that contains the letters '*a*' and '*b*'. Especially when considering that graph-based WSI can have its imperfections, meaning that sub-graphs could contain words from multiple different senses, it becomes extremely complex to normalize a sub-graph to a sense.

Secondly, graph-based methods have been used for disambiguation, but showed poor results when compared to topic-model WSI [5] and compared to other WSI methods [21, 16]. Based on these two reasons, it graph-based WSI is not employed for sense retrieval in this thesis.

2.3.3 Topic-model-based Approaches

Topic-model-based approaches aim to induce senses as topics. A topic model, latent Dirichlet allocation (LDA) or hierarchical Dirichlet process (HDP), is trained per abbreviation, resulting in topic probabilities for each occurrence of an abbreviation. Approaches deviated in how the topic probabilities are used to disambiguate. Chasin et al. [5], Brody and Lapata [4] and Goyal and Hovy [12] regarded each topic as a sense cluster, which meant that the occurrences were clustered based on their most probable topic. Chasin et al. mapped topics to senses by using annotated occurrences, such that they could conduct supervised evaluation. During their evaluation, they showed that HDP worked best in disambiguating abbreviations. LDA also performed better than the baseline of selecting the most frequent sense, while graph-based methods performed poorer than the baseline. Brody and Lapata, and Goyal and Hovy argued that topics can be represented as senses through the words that were most probable for each topic. Unfortunately, this presents the same problems as assigning senses to sub-graphs (see Section 2.3.2): a sense is inferred through a topic's most prominent words, which comes with uncertain sense labelling. Therefore, topic-model-based WSI are also not employed in this thesis.

2.3.4 Embedding-based Approaches

Embedding-based approaches utilize pre-trained language models to transform a word's context into a vectorized representation. Generally, the n context words are each represented as a d-dimensional embedding, and an aggregating function is used to reduce the dimensionality of the context representation from $n \cdot d$ to d. For example, Li et al. [19] proposed the usage of word embedding features over conventional features for disambiguating acronyms. The first embedding feature that they formulated was the surrounding-based embedding (SBE), which is the sum of word embeddings for words surrounding an acronym occurrence (see Equation 5). Here, $\texttt{SBE}(w_k)$ is the SBE for the kth word in a document, which is an acronym. $e(w_j)$ is the embedding of a word in the context of w_k, where the window size of the context is i.

$$\texttt{SBE}(w_k) = \sum_{j=k-i}^{k+i} e(w_j), j \neq k \tag{5}$$

The second embedding feature is the TF-IDF-based embedding (TBE). Where SBE uses the sum to aggregate over word embeddings, TBE first weighs each word embedding by the TF-IDF value of the context word relative to the documents that the acronym occurs in (see Equation 6). In $\texttt{TF-IDF}(w_j, D)$, D only includes documents that contain the acronym w_k.

$$\texttt{TBE}(w_k) = \sum_{j=k-i}^{k+i} \texttt{TF-IDF}(w_j, D) \cdot e(w_j), j \neq k \tag{6}$$

In their small scale experiment, Li et al. [19] used cosine similarity to disambiguate acronyms based on the surrounding-based embeddings for 5 words surrounding an acronym. SBE resulted in greater accuracy than TBE on two acronym datasets. SBE achieved lower accuracy than a conventional feature baseline on one dataset (93.10% and 95.29% respectively), but higher accuracy on the another dataset (94.86% and 74.29% respectively). Other baselines, including a majority sense baseline, performed far worse. It was argued that the conventional feature baseline performed well on the former dataset due to its low quality background ontology, compared to that of the latter dataset.

Wu et al. [41] were intrigued by the usage of word embeddings for WSD by Li et al. [19], so they trained their own word embeddings from a corpus containing 403,871 clinical notes. The word embeddings were generated using the neural network architecture by Collobert et al. [7], which used ranking lost criteria with negative sampling. They formulated three types of word embedding features: SBE as defined by Li et al., left-right SBE (LR_SBE) where the embeddings are aggregated separately for the right and left side of the abbreviation, and MAX_SBE where the maximum score for each embedding dimension of the surrounding words is taken. As a baseline, conventional features were TF-IDF weighted, which included: word features, word features with direction, positional word features, and word formation features for the abbreviation. The window size of surrounding words was 3 for both sides of the abbreviation, which was found to be optimal through 10-fold cross validation. Two clinical notes corpora containing annotated abbreviations, Vanderbilt University Hospital's (VUH) admission notes and University of Minnesota-affiliated (UMN) clinical notes, were used for training support vector machines for each type of features. The highest macro average accuracies were achieved using MAX_SBE (93.01% VUH, 95.79% UMN), while the baseline achieved the lowest accuracies (92.19% VUH, 94.97% UMN).

Jaber and Martínez [14] explored whether embeddings trained on a domain-specific vocabulary would be more suitable for clinical abbreviation disambiguation than embeddings trained on general English text in addition to domain-specific texts. They used an embedding pre-trained on biomedical articles, and another embedding that was pre-trained on Wikipedia articles and biomedical abstracts in addition to the biomedical articles. Their results showed that using the latter embedding increased WSD accuracy from 96.3% to 96.6% on average for two corpora and two different supervised classifiers. They concluded, similar to Wu et al. [41], that MAX_SBE result in better performance than averaging, taking the minimum or summing over surrounding word embeddings.

Similarly to Jaber and Martínez [14], Pakhomov et al. [29] experimented with using embeddings trained on: only clinical reports, only biomedical articles, and only Wikipedia articles. They tested the corpus domain effects of these embeddings on a semantic similarity and relatedness task for clinical terms. The embeddings trained on solely clinical reports and solely biomedical articles resulted in similar performance, while the embeddings trained on Wikipedia articles produced significantly worse results. Both Jaber and Martínez and Pakhomov et al. used word2vec with the skip-gram representations to generate word embeddings [24].

Based on the related work described in the previous paragraphs, it appears that embedding-based WSI results in better performance than feature-based WSI. The MAX_SBE vectorization method was empirically shown to be the best choice, so that will be employed in this thesis. Jaber and Martínez [14] and Pakhomov et al. [29] showed that the corpus domain of an embedding is important to NLP tasks. Therefore, MedRoBERTa.nl is used for generating embeddings in this thesis, since its corpus of Dutch medical reports is most similar to the healthcare reports considered in this thesis, compared to other currently available open-source models.

Substitution-based Approaches

Due to the rise of transformer models, often trained on a fill-mask objective, a new WSI method arose: substitution-based WSI. Substitution-based WSI entails predicting substitutes for the abbreviation in an occurrence. These substitutes are then clustered, just like any other WSI method, which allows for sense clustering.

Alagić et al. [1] utilized context2vec, a bidirectional LSTM based on word2vec's CBOW architecture, to extract lexical substitutes for words they wanted to disambiguate. They argued that lexical substitutes should be semantically very similar to the word they substitute, and allow for discriminating between a word's senses. Consider the example sentences containing the word "*play*" in Table 2. Their substitution-based WSI method entailed averaging over the embeddings of the 15 most suitable lexical substitutes for a word occurrence, and clustering the vectorized occurrences through affinity propagation. This method improved performance on several metrics with respect to approaches that use SBE, feature-based, graph-based, or topic-model-based approaches for the SemEval-2010 task [21].

Occurrence	**Sense**	**Substitutes**
The actor performed well in the **play**.	a dramatic work for the stage	*theater, performance, drama, musical*
The dog **play**s with a ball.	engage in activity for enjoyment	*performance, drama, musical, comedy*
The **play** at the local theater was amazing.	a dramatic work for the stage	*performance, drama, musical, stand-up*

TABLE 2: A fictional example depicting commonalities between senses and substitutes of the word "*play*".

Amrami and Goldberg [2] also utilized lexical substitutes for WSI, but they obtained these substitutes through an adapted fill-mask task. They split up the sentences for obtaining lexical substitutes at the position of the masked word. For example, the second sentence from Table 2 would be split up into: "*The dog* `<mask>`" and "`<mask>` *with a ball.*", where 10 substitutes would be used for each partial fill-mask task. Furthermore, they also experimented with adding the masked word together with "*and*" to each sentence part. In the example, this would look like: "*The dog* ***plays*** *and* `<mask>`" and "`<mask>` *and* ***plays*** *with a ball.*". They argued that including the masked word would result in better substitutes, especially when the other words in the sentence parts presented little information. Lastly, the substitutes would be TF-IDF weighted before being clustered. Agglomerative clustering was used, and the hard-clustering was transformed to soft-clustering such that the approach could be evaluated on the graded WSI task from SemEval-2013 [15]. On this task, their method significantly outperformed other methods for graded WSI.

Due to the state-of-the-art performance of substitution-based WSI (a subset of embedding-based WSI), it is employed in this thesis. Furthermore, MedRoBERTa.nl is also trained on the fill-mask objective, so it is suitable for substitution-based WSI.

2.4 Semantic Similarity for Abbreviation Normalization

Chopard and Spasić [6] were intrigued by WMD and its potential for disambiguating abbreviations. Since Kusner et al. [18] showed that relaxed WMD is almost as performant as WMD, while being far less computationally expensive, they used relaxed WMD for disambiguation. Their approach was to score candidate senses of an abbreviation based on the mean relaxed WMD of each sentence containing the candidate sense with respect to a sentence containing the abbreviation (see Equation 7). They referred to this score as the disambiguation score $\sigma(s_{abbr}, \phi)$. For a sentence containing an abbreviation, the surrounding words are represented in an embedding space: s_{abbr}. The same representation is used for every sentence containing a candidate long form: $s_\phi \in \mathrm{S}(\phi)$, where ϕ is a candidate sense from the set of all candidate senses for this abbreviation, Φ_{abbr}. The candidate sense with the lowest disambiguation score is considered to be the best one, i.e. $\phi^*_{s_{abbr}} = \min_{\phi \in \Phi_{abbr}} \sigma(s_{abbr}, \phi)$. This approach resulted in an accuracy of 96.36% in disambiguating abbreviations from the MSH WSD dataset, which contains abbreviations from biomedical abstracts. The next best approach on this dataset was from Li et al. [19], which disambiguated the abbreviations using a context vector representation and cosine similarity, and only obtained 95.29% accuracy.

$$\sigma(s_{abbr}, \phi) = \sum_{s_\phi \in \mathrm{S}(\phi)} \frac{\mathrm{WMD}(s_{abbr}, s_\phi)}{|\mathrm{S}(\phi)|} \tag{7}$$

Before disambiguating, it was necessary to extract candidate long forms for each abbreviation. For long form candidate extraction, a siamese RNN was trained using abbreviation-sense pairs from the CARD dataset [42]. One of the two RNNs that made up the siamese RNN was fed the encoding of an abbreviation, while the other RNN was fed the encoding of a n-gram that might be the sense of that abbreviation. The output of this siamese RNN was trained to be 1 if the abbreviation and sense were indeed a pair, and 0 if the sense was not a sense of the abbreviation. Negative samples were generated by taking an abbreviation and the sense of another abbreviation. This siamese RNN obtained a recall of 84.04% in recognizing correct abbreviation-sense pairs, compared to a simple pattern-matching-based baseline that had only 64.53% recall. Though the baseline had a precision of 90.12%, and the RNN 75.21%, it was argued that recall was more important than precision, since long forms that were not included in the candidate senses of a short form, would result in disambiguation errors.

Fascinated by this creative use of WMD, I explore the viability of candidate sense ranking using semantic similarity for sense retrieval in this thesis. If this works well, it can be employed to automatically retrieve abbreviation senses, or aid experts in their annotation effort.

3 Method

The method can be considered as two-fold: 1) comparing various WSI approaches to reduce the number of manual annotations needed to build a sense inventory, and 2) ranking candidate senses based on the semantic similarity of their occurrences with respect to the occurrences of abbreviations. The WSI approaches consist of transforming abbreviation occurrences to numerical vector representations (Section 3.1), and then clustering these vectors (Section 3.2). Figure 2 shows how these WSI clusters can be assigned senses by annotating their centroids. However, manual annotation could be reduced even more if it could be substituted by automatic annotation, which is where ranking candidate senses based on semantic similarity comes into play. The candidate senses should include any sense that an abbreviation might have, and can include irrelevant senses, but the number of candidate senses should be as small as possible. The occurrences of each abbreviation WSI cluster and the occurrences of each candidate sense are then compared using a semantic similarity metric. This allows for ranking the candidate senses with respect to each abbreviation WSI cluster, such that the highest ranking candidate sense should be the sense of the occurrences in the cluster.

The complete method is depicted in Figure 3. The candidate sense that is most similar to the cluster is likely the sense of that cluster. In the Figure, this was the candidate sense with the lowest mean WMD distance: "*antibiotics*". Note that the example only shows what happens for perfect WSI and perfect similarity ranking. In practice, the correct sense should rarely rank highest due to the large number of candidate senses.

3.1 Occurrence Vectorization

Three methods were used to transform abbreviation occurrences into vector representations: 1) pmi-weighted conventional features (similar to Xu et al. [45]), 2) MAX_SBE using the MedRoBERTa embeddings from words in the context of an abbreviation, and 3) pmi-weighted substitution words using MedRoBERTa. Topic models and graph-based methods were excluded, since they had shown to perform more poorly than other WSI approaches [21, 15, 5, 2], and would be difficult to retrieve senses for. For the conventional features and MAX_SBE vectorization, the size of the context window was set to 10, since that was the smallest context size that showed peak performance on WSD by Moon et al. [25]. The number of substitutes was set to 20, which was the same as Amrami and Goldberg [2].

Firstly, the conventional features consist of three types of features, which were pmi-weighted, similar to Xu et al. [45]. Rather than being stemmed, the context words were lemmatized. They are shown below together with examples based on the example report shown in Table 3.

Feature 1 The lemmatized context words surrounding an abbreviation **without** positional information: [*hb*, *de*, *client*, *was*, *incontinent*, *vanwege*, *infectie*, *ab*, *toedienen*, *daarnaast*, *nog*, *navragen*, *over*, *weekend*, *plan*, *hij*, *zeggen*]

Feature 2 The lemmatized context words surrounding an abbreviation **with** positional information: [*L7_hb*, *L6_de*, *L5_client*, *L4_was*, *L3_incontinent*, *L2_vanwege*, *L1_infectie*, *R1_ab*, *R2_toedienen*, *R3_daarnaast*, *R4_nog*, *R5_navragen*, *R6_over*, *R7_*

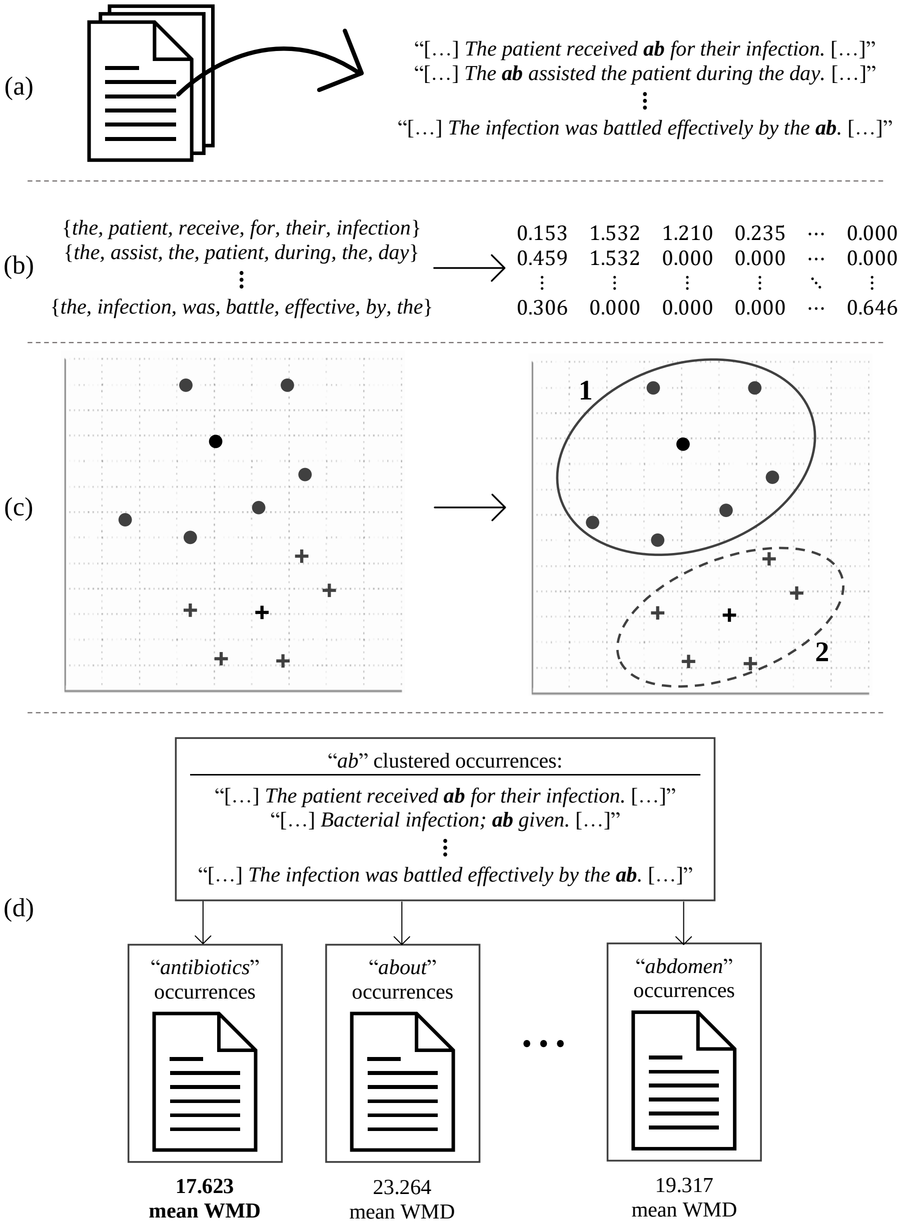
(a)
"[…] The patient received **ab** for their infection. […]"
"[…] The **ab** assisted the patient during the day. […]"
⋮
"[…] The infection was battled effectively by the **ab**. […]"
(b)
{the, patient, receive, for, their, infection}
{the, assist, the, patient, during, the, day}
⋮
{the, infection, was, battle, effective, by, the}
0.153 1.532 1.210 0.235 ⋯ 0.000
0.459 1.532 0.000 0.000 ⋯ 0.000
⋮ ⋮ ⋮ ⋮ ⋱ ⋮
0.306 0.000 0.000 0.000 ⋯ 0.646
(c)
1
2
(d)
"ab" clustered occurrences:
"[…] The patient received **ab** for their infection. […]"
"[…] Bacterial infection; **ab** given. […]"
⋮
"[…] The infection was battled effectively by the **ab**. […]"
"antibiotics" occurrences
"about" occurrences
"abdomen" occurrences
17.623 mean WMD
23.264 mean WMD
19.317 mean WMD

FIGURE 3: The method for extracting abbreviation senses automatically using WSI and semantic similarity ranking of candidate senses. In step (a), the occurrences of an abbreviation were extracted from healthcare reports. In step (b), the context of an abbreviation was transformed into a vector for each occurrence (the vectorization methods are described in Section 3.1). In step (c), the occurrence vectors were clustered together, such that a WSI cluster would only contain occurrences of the same sense. In step (d), the occurrences in each abbreviation WSI cluster were compared to occurrences of candidate senses in terms of semantic similarity.

weekend, *R8_plan*, *R9_hij*, *R10_zeggen*] - ‘*L*’ and ‘*R*’ indicate whether the context word occurs to the left or right of the abbreviation, and the number indicates how far to the right or left.

Feature 3 The report type, such as fluid intake, medicine, et cetera: [*REPORT_TYPE_rapportage*]

Secondly, MAX_SBE was used as aggregation over context word embeddings, since it showed better results than other surrounding-based embeddings [41, 14]. The embeddings for context words were found by passing an abbreviation occurrence to MedRoBERTa where the abbreviation would be masked. The embedding of the masked abbreviation was removed from the matrix of word embeddings. Then, the word embeddings were aggregated over each embedding dimension by taking the maximum (see Equation 8). Here, w_k was the abbreviation in the occurrence for which the MAX_SBE needs to be calculated. w_j is a word in the context of w_k within a context window of size c.

$$\texttt{MAX\_SBE}(w_k) = \sum_{j=k-c}^{k+c} e(w_j) \qquad\qquad j \neq k, \quad w_j \notin \{\text{‘<}\textit{mask}\text{>’}, \text{‘.’}, ...\} \tag{8}$$

The example shown in Table 3 would first be tokenized by MedRoBERTa.nl’s WordPiece tokenizer. The tokens that would be used for calculating the MAX_SBE would be: [*hb*, *De*, *cliënt*, *was*, *incontinent*, *vanwege*, *infectie*, *ab*, *toegediend*, *Daarnaast*, *nog*, *nagevraagd*, *over*, *weekend*, *plannen*, *Hij*, *zei*]. The result per abbreviation occurrence would be a vector that has the same number of dimensions as the embeddings from MedRoBERTa.nl.

Lastly, Amrami and Goldberg [2], Alagić et al. [1] showed that lexical substitutes work well for WSI. Furthermore, this occurrence vectorization method suited MedRoBERTa, since RoBERTa models were trained on the fill-mask task. Rather than using a context window surrounding an abbreviation, only the sentence that the abbreviation occurred in would be tokenized and used to predict substitutes for the masked abbreviation. The top-20 predicted substitutes for “*wv*” in the example shown in Table 3 would be: [*en*, *waarvoor*, *wv*, *dus*, *met*, *heeft*, *geen*, *daarom*, *is*, *kreeg*, *zonder*, *na*, *voor*, *waarop*, *van*, *waardoor*, *werd*, +, *toen*]. The substitutes for all occurrences of an abbreviation were lemmatized and then pmi-weighted.

3.2 Sense Clustering

Two methods for clustering were used: 1) Tight Clustering for Rare Senses (TCRS) [45], and 2) k-means clustering. The motivation behind selecting TCRS was that it was designed with the purpose of discovering rare senses. The reason for using k-means + Gap was that it was used by several approaches during SemEval-2010 [21], an evaluation task for WSI, and showed good results. Furthermore, k-means was a relatively intuitive centroid-based clustering algorithm, while TCRS was computationally more intensive and a density-based clustering algorithm.

TCRS required values for its similarity threshold parameters, which were optimized for high sense coverage versus low annotation cost by Xu et al. [45]. The number of annotations for creating the dataset (see Section 4) was already limited, so this optimization

	Report text	Abbreviation senses

Report (Dutch)	hb. De cliënt was incontinent vanwege infectie **wv** ab toegediend. Daarnaast nog nagevraagd over weekend plannen. Hij zei van plan te zijn de bingo op vrijdag bij te wonen. Verder gb	hb = huisbezoek;
		wv = waarvoor;
		ab = antibiotica;
		gb = geen bijzonderheden.

Report (translated)	hv. The client has lost bladder control due to an infection **fw** ab was prescribed. Also asked about weekend plans. He responded to plan on joining the Friday afternoon bingo. Besides that nr	hv = house visitation;
		fw = for which;
		ab = antibiotics;
		nr = no remarks.

TABLE 3: A fictional Dutch healthcare report that captures the usage of abbreviations.

step was not adopted. Instead, two configurations of TCRS were used for sense clustering. The first configuration used the values for the similarity thresholds as they were found to be optimal by Xu et al. The second configuration defined the similarity thresholds based on the distribution of similarities between occurrences, such that the similarity threshold for tight clustering was the 99th percentile from occurrence similarities, and the similarity threshold for merging clusters using complete linkage was the 25th percentile from occurrence similarities. These heuristically determined thresholds were based on that tight clusters should only contain the occurrences that were most similar, and complete linkage should be allowed for reasonably dissimilar occurrences.

The number of clusters for k-means was decided through the Gap-statistic [35]. The Gap statistic is the difference between the within-cluster dispersion of a clustering to that of a null reference distribution based on the data. As suggested by Tibshirani et al. [35], the number of clusters k was chosen to be the smallest k such that $Gap(k) \geq Gap(k+1) - s_{k+1}$. Here, $Gap(k)$ was the average Gap value when using k number of clusters, and s_{k+1} was the standard deviation of $Gap(k+1)$. Furthermore, two fixed values for k were selected: a small $k = 10$ and a larger $k = 20$.

3.3 Semantic Similarity Ranking

Aside from assigning a sense to a cluster based on its centroid, two methods were constructed for ranking candidate senses based on the semantic similarity of their occurrences to occurrences contained within an abbreviation cluster: 1) ranking based on the lowest relaxed WMD, and 2) ranking based on the highest cosine similarity of MedRoBERTa.nl sentence embeddings. The candidate senses of an abbreviation were:

- each n-gram that occurred 10 or more times in the corpus, where n is between 1 and the number of characters in the abbreviation;
- which contained the letters of an abbreviation in consecutive order;
- and was not the abbreviation itself (with or without additional punctuation or capitalization).

For example, candidate senses for the abbreviation "*ab*" include: "*antibiotica (antibiotics)*", "*aardbei (strawberry)*" and "*aan begeleider (to counselor)*". However, the candidate senses do not include: "*a.b.*", "*AB*" or any other syntactical variants of the abbreviation itself.

The first ranking method was based on relaxed WMD (see Section 2.4). Candidate senses were ranked based on their semantic distance score, $\sigma(\alpha, \phi)$, i.e. the average relaxed WMD (see Equation 9). For each sentence in a cluster containing an abbreviation s_α, the relaxed WMD was calculated with respect to each sentence containing a candidate sense s_ϕ. These distances were summed and normalized over the number of sentences in a cluster $|\texttt{S}(\alpha)|$ and the number of sentences containing a candidate sense $|\texttt{S}(\phi)|$. There would thus be a candidate sense ranking per cluster of size $|\texttt{S}(\phi)|$, where the top ranking candidate sense had the lowest semantic distance score.

$$\sigma(\alpha, \phi) = \sum_{s_\alpha \in \texttt{S}(\alpha)} \sum_{s_\phi \in \texttt{S}(\phi)} \frac{\texttt{relaxedWMD}(s_\alpha, s_\phi)}{|\texttt{S}(\alpha)| \times |\texttt{S}(\phi)|} \tag{9}$$

The second ranking method was based on the cosine similarity between sentence embeddings obtained through MedRoBERTa.nl. A sentence embedding was obtained by taking the sum over the wordpiece embeddings of that sentence (see Equation 10). Since cosine similarity was used to compare sentence embeddings, the similarity values would remain the same as when the mean over the wordpiece embeddings was taken instead of the sum.[2] Equation 12 shows how the similarity score for an abbreviation cluster and candidate sense was computed. The candidate sense with the highest similarity score was ranked highest.

$$e(s) = \sum_{w \in s} e(w) \tag{10}$$

$$\texttt{cosSim}(s_1, s_2) = \frac{e(s_1) \cdot e(s_1)}{||e(s_1)||_2 \times ||e(s_2)||_2} \tag{11}$$

$$\texttt{sim}_{\texttt{score}}(\alpha, \phi) = \sum_{s_\alpha \in \texttt{S}(\alpha)} \sum_{s_\phi \in \texttt{S}(\phi)} \frac{\texttt{cosSim}(s_\alpha, s\phi)}{|\texttt{S}(\alpha)| \times |\texttt{S}(\phi)|} \tag{12}$$

In this thesis, the candidate sense ranking was not used to build a sense inventory, since it was first necessary to see whether it works. If the sense of an abbreviation is often included in the candidates and ranks high, then candidate sense ranking can be used to build a sense inventory. For instance, the top-20 candidate senses can be supplied to experts during annotation, if the sense of an abbreviation is often listed in this top-20.

[2] This holds in theory, but due to rounding errors in floating-point arithmetic, slight differences could occur. It is better to use the sum, since this refrains from making an unnecessary additional computation.

4 Dataset

Similar to Xu et al. [45], a dataset was created by annotating occurrences of abbreviations in healthcare reports. This dataset functions as a gold standard for the various senses that an abbreviation can have, and an estimation of the sense frequencies. The size of the dataset was limited by the availability of expert annotators. Therefore, the abbreviations for this dataset were selected based on the following criteria:

1. They should consist of two letters, since those abbreviations are statistically more likely to have multiple senses than longer abbreviations, based on the sense inventory containing English clinical abbreviations by Moon et al. [26]. 22.5% of that sense inventory consists of two-letter abbreviations, which have 53.9% of the senses.

2. They should occur at least 1000 times within 1,000,000 reports of their domain, since their more frequent usage makes them more interesting than less frequently used abbreviations.

3. They should be used by at least eight out of ten healthcare providers within the domain. This indicates the abbreviations were domain-specific, rather than provider-, employee-, or client-specific.

4. They should be indicative of having multiple domain-specific senses, since abbreviations that could be ambiguous in a clinical context were shown to be problematic in Section 1.

Healthcare reports from Nedap Healthcare were used to identify abbreviations that meet the criteria stipulated above. The reports originate from the three healthcare domains where Nedap Healthcare provides administrative healthcare solutions: elderly care, mental healthcare, and disability care.

100,000 reports were tokenized for ten providers of each healthcare domain, totaling 1,000,000 reports per domain. The reports were deidentified, i.e. stripped from protected health information, using a tool developed in a previous thesis at Nedap [36]. The providers were those with the largest number of clients and more than 100,000 available reports. They opted in for making their reports available to Nedap for product improvement. A heuristic was used to identify two-letter tokens that occur in the syntax of an abbreviation (see Appendix A). This heuristic allows for finding abbreviations that fit criteria 1-3. The occurrences of each remaining abbreviation were clustered using the WSI method from Xu et al. [45], which is described in Section 3, since that method showed to work well on WSI for English clinical abbreviations. Then, I identified which abbreviations certainly had multiple domain-specific senses by inspecting cluster centroids as a non-expert. As a result, the abbreviations in Table 4 fit the criteria stipulated above. From criterion 1 to criterion 4, the diminishing number of abbreviations is shown in Table 5 per healthcare domain.

Though the number of abbreviations is small, other studies also used few words or abbreviations for WSI: 14 English clinical abbreviations [45]; 75 English clinical abbreviations; 50 English nouns and 50 verbs [21]; 20 English nouns, 20 verbs, and 10 adjectives [15]. For my thesis, the resources were simply not available to increase the size of the

Elderly care	Mental healthcare	Disabled care
AB	BW	AB
DD	CM	DD
HB	GG	GB
HH	GV	HB
WV	GZ	HH
	SU	IB

TABLE 4: The abbreviations that fit the four criteria specified in Section 4 for the elderly care, mental healthcare, and disabled care sectors.

Criterion	Elderly care	Mental healthcare	Disabled care	Total
None	1003	1947	1255	4205
1	307	430	315	1052
2	46	77	31	154
3	33	67	29	129
4	5	6	6	17

TABLE 5: This table displays the number of abbreviations that meet the criteria to the left cumulatively. For example, the number '77' at the center of the table indicates that 77 unique tokens are abbreviations that fit criteria 1 and 2. *None* indicates no criteria are used to filter the tokens, so this denotes the number of unique tokens that appear in an abbreviation syntax.

dataset. I mitigate this resource issue by using a creative experimental set-up (Section 5) and performing an extensive error analysis (Section 6.2).

4.1 Data Sampling

For each of the abbreviations, 60 occurrences were sampled. This sample size ensures that, for an abbreviation that has two senses, a 0.05 frequent minority sense has a probability of more than 0.95 to be included within the sample. In other words, it would be improbable that all 60 occurrences of a homonymous abbreviation would have the same sense. Furthermore, the sample size ensured a reasonable workload for each domain expert: roughly 2 hours of annotation.

To sample from the abbreviation occurrences, the occurrences were deduplicated and binned based on context overlap. The context of an occurrence consisted of the five tokens preceding and five tokens succeeding an abbreviation. Occurrences for which the context overlapped completely were considered duplicates, and occurrences that had three or more overlapping tokens were binned together. The bins were sampled with a probability relative to the size of the bin, which ensures that the frequency of contextually similar occurrences are the same in the sample as in the whole dataset. Since an abbreviation's context and its sense are hypothesized to be closely related (Section 2.3), the bin sampling ensures that the frequency of senses in the sample is similar to the frequency of senses in the whole dataset.

4.2 Annotation Process

An expert per healthcare domain annotated the 60 sampled occurrences per abbreviation of that domain. The elderly care expert was a general practitioner. The mental healthcare expert was a healthcare psychologist (the exact job title is '*gezondheidszorgpsycholoog*' in Dutch), who was in full-time practice up to a year ago. Lastly, the disability care expert was a healthcare nurse with several years of experience.

Four distinct scenarios were identified for annotating an abbreviation occurrence:

1. The abbreviation in the occurrence was annotated with a single sense (long form).
2. The abbreviation in the occurrence was annotated with multiple senses, meaning that the annotator believes that the abbreviation has one of those senses.
3. The token is not an abbreviation in this occurrence, such as postal codes. I excluded most of these occurrences from the sample of 60 occurrences by inspecting them beforehand.
4. The abbreviation cannot be annotated with a sense in this occurrence, either due to a lack of context or a lack of knowledge by the expert.

Table 6 and Table 7 show the abbreviations, their senses, their senses translated to English, and sense frequencies relative to the abbreviation. Unexpectedly, the abbreviations "*gv*" and "*su*" from the mental healthcare sector only showed to have a single sense. Furthermore, the abbreviation "*ab*" showed to have a completely different sense frequency

Domain	**Abbr-eviation**	**Sense**	**Translated sense**	**Relative frequency**
elderly care	ab	antibiotica	antibiotics	0.966
		activiteitenbegeleider	activity counselor	0.034
	dd	de die	per day	0.442
		de dato	dated (of)	0.212
		dagdienst	day shift	0.192
		dienstdoende	on duty	0.077
		differentiaaldiagnose	differential diagnosis	0.077
	hb	hemoglobine	hemoglobin	0.929
		huisbezoek	house visitation	0.071
	hh	huishoudelijke hulp	domestic assistance	0.673
		herhalingen	repetitions	0.309
		hoofd hals	head neck	0.018
	wv	waarvoor	for which	0.839
		wijkverpleegkundige	district nurse	0.161
mental healthcare	bw	beschermd wonen	protected living	0.983
		bewindvoerder	curator	0.017
	cm	contactmoment	contact moment	0.924
		casemanager	casemanager	0.057
		centimeter	centimeter	0.019
	gg	groepsgenoot	group mate	0.846
		geen gehoor	no answer	0.091
		gegevens	information	0.036
		gegeven	given	0.027
	gv	gezonde volwassene	healthy adult	1.000
	gz	gezondheidszorg(psycholoog)	healthcare (psychologist)	0.846
		gezonde volwassene	healthy adult	0.128
		gezins	family	0.026
	su	suicide	suicidal	1.000

TABLE 6: List of abbreviations per healthcare domain and their senses (part 1/2). The abbreviations are sorted in alphabetical order, and their senses are sorted on relative sense frequency.

Domain	**Abbr- eviation**	**Sense**	**Translated sense**	**Relative frequency**
disability care	ab	activiteitenbegeleiding	activity counseling	0.710
		ambulant begeleider	ambulatory assistant	0.161
		antibiotica	antibiotics	0.129
	dd	de die	per day	0.463
		de dato	dated (of)	0.296
		differentiaaldiagnose	differential diagnosis	0.167
		Donald Duck	Donald Duck	0.037
		dienstdoende	on duty	0.037
	gb	geen bijzonderheden	no remarks	0.583
		gezinsbegeleider	family counselor	0.292
		gigabyte	gigabyte	0.125
	hb	huisbezoek	house visitation	0.927
		hemoglobine	hemoglobin	0.073
	hh	huishoudelijke hulp	domestic assistance	0.659
		herhalingen	repetitions	0.341
	ib	individuele begeleider	individual counselor	0.962
		intern begeleider	internal counselor	0.038

TABLE 7: List of abbreviations per healthcare domain and their senses (part 2/2).

distribution for elderly care compared to disabled care, which exemplifies the need for retrieving senses per domain.

Contrary to Moon et al. [26], who did not report any occurrences that the annotators were unable to annotate, Table 8 shows that annotators were unable to annotate about 16.4% of abbreviation occurrences. When an abbreviation was not annotated by a sense, it was 53.4% due to a lack of knowledge by the annotator, 39.5% due to a lack of context, and 7.1% due to the occurrence not containing an abbreviation.

The expert on elderly care also annotated 10 occurrences for each of the disability care abbreviations, such that inter-annotator agreement could be measured. As a general practitioner, this expert has some expertise on disability care. From the total of 60 occurrences for inter-annotator agreement, only 26 occurrences were annotated by both annotators. However, the annotators completely agreed on the senses of these 26 occurrences.

Domain	**Annotated**	**Lacking context**	**Non-abbreviation**	**Unfamiliar**
Elderly care	55.6 ± 2.2	2.2 ± 2.9	0.4 ± 0.5	1.8 ± 1.6
Mental healthcare	53.7 ± 6.9	1.0 ± 1.2	1.5 ± 1.0	3.8 ± 6.4
Disability care	41.2 ± 10.9	8.5 ± 8.1	0.2 ± 0.4	10.2 ± 11.0

TABLE 8: Frequency of annotation scenarios for each healthcare domain.

5 Experiments

Two experiments were conducted to evaluate the methods pictured in Figure 3: 1) each combination of vectorization and clustering, which forms a WSI approach, was used to cluster abbreviation occurrences; and 2) a ranking of candidate senses was made for each abbreviation based on annotated occurrences and clusters from the most suitable WSI approach. The former experiment used the methods described in Section 3.1 and Section 3.2, while the latter experiment used the methods described in Section 3.3.

5.1 Experiment 1: Word-Sense Induction

The objective of the first experiment was to answer subRQ-1 and subRQ-2. SubRQ-1 regards reducing annotation cost through WSI compared to annotating a random sample of abbreviation occurrences. SubRQ-2 regards improving sense coverage through WSI compared to annotating a random sample of abbreviation occurrences. The experiment was conducted for the abbreviations listed in Section 4 except for the abbreviations with only a single sense, which were the mental healthcare abbreviations “*gv*” and “*su*”. Including those abbreviations would have made it appear as if the WSI methods had a higher sense coverage than that they can actually obtain for homogeneous abbreviations, since the sense coverage for these single-sense abbreviation would be 1, regardless of clustering. Of course, there would be many healthcare abbreviations in the real world that only have a single sense, but it would be left to whomever wants to build a sense inventory whether they want to use WSI for retrieving an abbreviations senses. If they wanted to know the sense of an 8 letter abbreviation, it would be likely that this abbreviation would only have a single sense across all its occurrences.

The same 100,000 reports per healthcare provider per healthcare sector that were used for the ground truth dataset were used for WSI. First, any protected health information was replaced with a surrogate through deidentification [36]. For example, the name “*Walter White*” would be replaced with “*PERSON*”. Secondly, the reports were processed by a SpaCy v3.4 NLP pipeline trained on Dutch news articles and web page texts. This pipeline included tokenization, lemmatization and sentence segmentation. Thirdly, each token that consisted of the letters that make up an abbreviation, regardless of casing and optionally including periods, was considered an occurrence of that abbreviation. So, “*AB*”, “*a.b.*” and “*ab*” would all be considered occurrences of the abbreviation “*ab*”. Fourthly, the abbreviation occurrences were deduplicated. An occurrence was a duplicate, if it had the exact same context for a window of 20 words as another abbreviation occurrence. Lastly, to conclude step (a) from Figure 3, a sample of 1000 occurrences was taken per abbreviation, which included the labelled occurrences. This sample size was also used by Xu et al. [45], and allowed them to cluster occurrences of rare ($< 2\%$ frequent) senses.

An abbreviation’s occurrences were vectorized in the three different ways described in Section 3.1: as pmi-weighted conventional features, as MAX_SBE, and as pmi-weighted substitution lemmas. After that, they were clustered in the 5 different ways described in Section 3.2: TCRS with percentile-based similarity thresholds, TCRS with fixed similarity thresholds, k-means with the Gap statistic, k-means with $k = 10$, and k-means with $k = 20$. It would have required too many annotations to manually annotate each cluster centroid (depicted in Figure 2), since this would have to be done for each WSI method and for each repetition of the experiment. Therefore, clusters were assigned a sense based on the few annotated occurrences (40-60 out of 1000, depending on how many the experts could annotate). For clusters that contained annotated occurrences, a cluster was assigned the sense of the annotated occurrence in the cluster that lied closest to the cluster center. There were also clusters without any annotated occurrences in them, which were assigned a sense based on the sense frequencies listed in Section 4. This cluster sense assignment based on the limited number of annotated occurrences is depicted in Figure 4. The whole experiment from sampling onwards was repeated 10 times for each vectorization-clustering combination to reduce the effect of randomness.

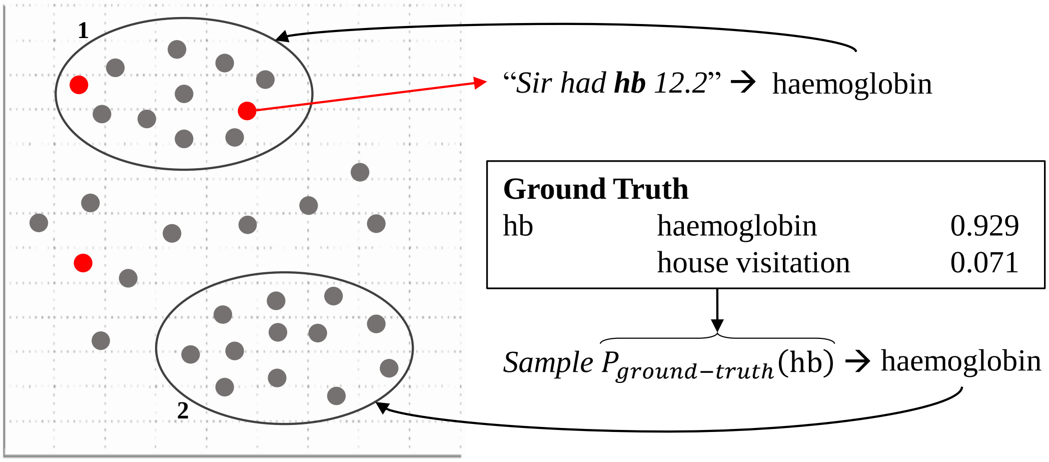
1
"*Sir had **hb** 12.2*" → haemoglobin
Ground Truth
hb
haemoglobin 0.929
house visitation 0.071
Sample $\overbrace{P_{ground-truth}(\text{hb})}$ → haemoglobin
2

FIGURE 4: This picture shows a clustering of vectorized abbreviation occurrences on the left side. The red dots are annotated occurrences, while the gray dots are not annotated. The points within an ellipse from a cluster, and the datapoints outside the ellipses are considered noise. Note that only TCRS can regard datapoints as noise, while k-means clustering cannot. Two scenarios for assigning a sense to a cluster are displayed. Cluster-1 is assigned a sense based on the annotated occurrence closest to its center. Cluster-2 contains no annotated occurrences, and is therefore assigned a sense based on the sense frequency distribution from the ground truth.

5.1.1 WSI Evaluation Measures

Sense coverage and annotation cost are the relevant metrics for WSI with respect to subRQ-1 and subRQ-2. Sense coverage (see Equation 13) is measured as the fraction of senses identified through WSI ($|\Phi_{\text{WSI}}|$) from the senses in the gold standard ($|\Phi_{\text{gs}}|$), exactly as done by Xu et al. [45]. Here, Φ represents a set of senses. The annotation cost (see Equation 14) is measured differently from Xu et al.. Annotation cost is measured as the number of clusters found through WSI ($|C_{\text{WSI}}|$), and not relative to the number of annotations used to create the gold standard. Here, C is a set of abbreviation clusters, since one annotation is necessary to find the sense of each cluster.

$$\text{sense coverage}(\text{WSI}) = \frac{|\Phi_{\text{WSI}}|}{|\Phi_{\text{gs}}|} \tag{13}$$

$$\text{annotation cost(WSI)} = |C_{\text{WSI}}| \tag{14}$$

The sense coverage and annotation cost are also calculated for annotating randomly sampled abbreviation occurrences, which will be referred to as the *baseline*. For the baseline, the sense coverage is still the fraction of identified senses over gold standard senses, and the annotation cost is the sample size of the baseline. Since the sense coverage can be calculated for both a WSI method and the baseline, the *sense coverage gain* can be calculated (see Equation 15). Similarly, the *annotation cost reduction* can be calculated (see Equation 16).

$$\text{sense coverage gain} = \frac{\text{sense coverage(WSI)} - \text{sense coverage(baseline)}}{\text{sense coverage(baseline)}} \tag{15}$$

$$\text{annotation cost reduction} = \frac{\text{annotation cost(baseline)} - \text{annotation cost(WSI)}}{\text{annotation cost(baseline)}} \tag{16}$$

Rather than showing sense coverage, annotation cost in a large table, these metrics are plotted as exemplified in Figure 5. The baseline is depicted as a line plot, where each point on the line plot shows the sense coverage that is obtained by annotating a number of randomly sampled occurrences. A WSI method's sense coverage and annotation cost are depicted similar to a 2-dimensional box plot. The dot inside the ellipse is the mean annotation cost and sense coverage for a WSI method (annotated with text in the figure for clarity). The ellipse shows the first and third quartiles of the annotation cost and sense coverage in the shape of an ellipse rather than a box.

This visualization (Figure 5) allows the reader to easily observe the performance, variance and skewness of a WSI method in terms of annotation cost and sense coverage. Furthermore, the sense coverage gain can be observed by projecting the mean sense coverage of a WSI method on the baseline (in this example visualized by a vertical orange line), and the annotation cost reduction can be observed similarly (in this example visualized by a horizontal orange line). Therefore, any WSI method that has a mean to the top left of the baseline reduces annotation cost and improves sense coverages with respect to the baseline. Equivalently, any WSI method that has a mean to the bottom right of the baseline is worse than the baseline. The results will also include a table containing the annotation cost reduction and sense coverage gain in addition to the plots.

5.2 Experiment 2: Ranking Candidate Senses

The objective of the second experiment was to answer subRQ-3, which regards ranking candidate senses using semantic similarity measures. This experiment was conducted in two parts: 1) ranking candidate senses based on their similarity to annotated occurrences of a single sense, and 2) ranking candidate senses based on WSI clusters. The former allowed for evaluating the effectiveness of semantic similarity ranking without influence from how well-formed the WSI clusters were. The latter allowed for evaluating the effectiveness of the entire method, as depicted in Figure 3.

The ranking using annotated occurrences was executed for every sense in the ground truth that also occurred more than 10 times within the reports. This was not the case

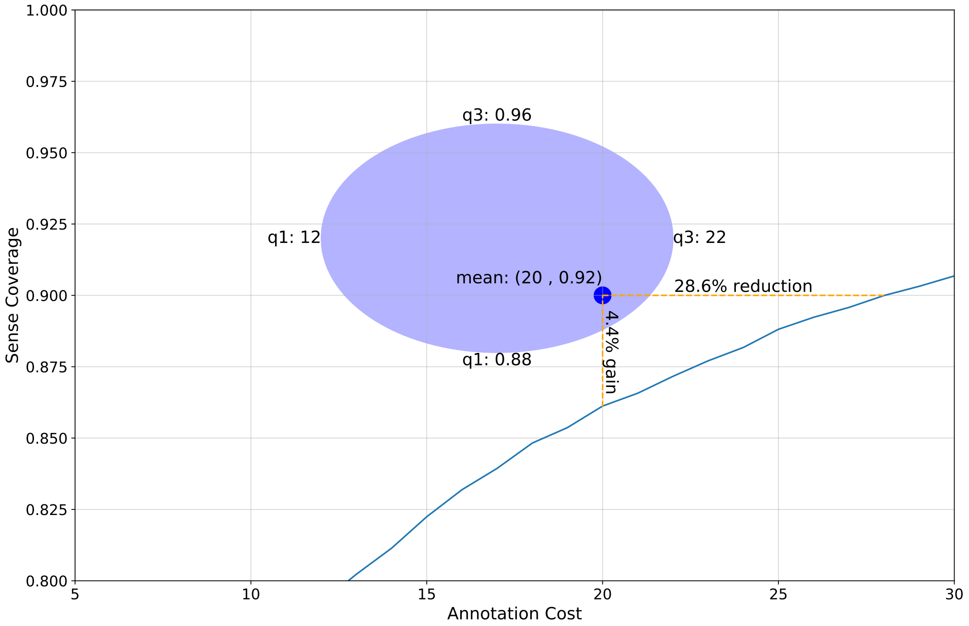

FIGURE 5: Plot that visualizes how a WSI method performs with respect to a baseline in terms of sense coverage and annotation cost. The baseline, which entails annotating randomly sampled abbreviation occurrences, is depicted as a blue line plot. The sense coverage and annotation cost of a WSI method is depicted as a blue dot and ellipse. The dot depicts the mean annotation cost and sense coverage of the WSI method, which is annotated with 'mean: (20, 0.92)' for clarification in this example. The ellipse is the box of a 2-dimensional box plot, where the left end of the ellipse is the first quartile of annotation cost, which is annotated with 'q1: 12'. Similarly, the right end is the third quartile of annotation cost ('q3: 22'), the bottom end is the first quartile of sense coverage ('q1: 0.88'), and the top end is the third quartile of sense coverage ('q3: 0.88'). To exemplify how a WSI method can be compared to the baseline, a horizontal dashed orange line is shown for annotation cost reduction ('28.6% reduction'), and a vertical dashed orange line for sense coverage gain ('4.4% gain').

for several senses, so these were excluded from the ranking evaluation. The ranking was based on sentence similarity and relaxed WMD, as described in Section 3.3.

The ranking using WSI clusters was only conducted for the WSI approach with the highest V-measure, which is the harmonic mean of homogeneity and completeness. Homogeneity and completeness were calculated based on the annotated occurrences residing in each cluster. As mentioned before, TCRS regards some occurrences as noise, which requires adapted definitions for homogeneity and completeness for TCRS. The homogeneity of a TCRS clustering is calculated only for the clusters without noise. The completeness of a TCRS clustering is calculated for all non-noise cluster, as well as one-cluster-per-occurrence clusters for each annotated occurrence that is considered noise. If the completeness would only be calculated for the non-noise clusters instead, a TCRS clustering of only a single non-noise cluster containing a single annotated occurrence would result in a homogeneity of 1 and a completeness of 1. The adapted completeness would be 0 for this clustering.

High homogeneity means that occurrences within a cluster are largely of the same sense, which is necessary for the semantic similarity ranking to work. An inhomogeneous cluster would not have one correct candidate sense that is relevant to most occurrences of the cluster, which means that candidate sense ranking would not work. High completeness means that occurrences of the same sense are located among a few clusters, rather than being spread out over many clusters. Clusters with too few or a single occurrence would have a greater chance of coincidentally being similar to an irrelevant candidate sense. For example, the sense of "*ab*" in the sentence "*The urologist recommended ab for the bladder infection.*" is "*antibiotics*", but due to the urology-related context, the highest ranking candidate sense for the sentence could be "*abdominal bladder*". If the sentence is clustered together with many more sentences of the sense "*antibiotics*" that did not have the urology-related context, the incorrect candidate sense "*abdominal bladder*" would rank lower and the correct candidate sense "*antibiotics*" would rank higher.

The homogeneity and completeness will be shown in a scatter plot for each WSI method. This allows for easier observation on how each WSI method relates to another in terms of homogeneity and completeness, than a table would allow. The V-measures will still be shown in a table.

5.2.1 Candidate Sense Ranking Evaluation Measures

The candidate senses were ranked using annotated occurrences, and using the clusters of the most suitable WSI method in terms of V-measure. This ranking is evaluated using the mean reciprocal rank (MRR) and mean rank, where only a single sense in the ranking is considered as relevant. For the candidate sense ranking using annotated occurrences of the same sense, the exact sense (so no inflections) of the occurrences is considered the only relevant result in ranking. For the candidate sense ranking using a cluster of occurrences of the same abbreviation, the highest ranking sense of that abbreviation (based on the senses contained in the ground truth dataset) is only considered relevant.

As a baseline, the candidate senses are ranked arbitrarily. The rank of a sense in an arbitrary ranking of length n can be modelled as a random variable that has a discrete uniform distribution: $X \sim \texttt{Unif}(1, n+1)$. Therefore, the expected rank of the baseline is $\frac{n}{2}+1$ in context of the ranking using annotated occurrences, since only a single sense in the ranking is correct. Similarly, the baseline expected rank can be modelled in context of the ranking using clustered occurrences, since multiple senses can define the highest rank in the ranking. The rank of the highest ranking sense out of m senses can be modelled as shown in Equations 17 through 24. Therefore, the expected rank of the baseline is $\frac{n}{m+1} + 1$.

$$X_1, X_2, \ldots, X_m \sim \texttt{Unif}(1, n+1) \qquad \text{i.i.d.} \tag{17}$$

$$Y = \min\{X_1, X_2, \ldots, X_m\} \tag{18}$$

$$\begin{aligned} F(y) &= P(\min\{X_1, X_2, \ldots, X_m\} \geq y) && (19) \\ &= P(X_1 \geq y) \cdot P(X_2 \geq y) \cdots P(X_m \geq y) && (20) \\ &= \left(\frac{y-1}{n}\right)^m && (21) \end{aligned}$$

$$\begin{aligned} \mathrm{E}(Y) &= \int_0^1 (1 - F(y))dy + \int_1^n (1 - F(y))dy + \int_n^{\text{inf}} (1 - F(y))dy && (22) \\ &= 1dy + \int_1^n \left(1 - \left(\frac{y-1}{n}\right)^m\right) dy + 0dy && (23) \\ &= 1 + \frac{n}{m+1} && (24) \end{aligned}$$

6 Evaluation

This section includes the results of the two experiments stipulated in Section 5. The results are supplemented by an extensive error analysis.

6.1 Results

Figure 6, 7, and 8 show a box-plot-like visualization of annotation cost versus sense coverage for each WSI method plotted together with the baseline. Each plot shows 5 WSI methods that all use the same vectorization: Figure 6 displays WSI methods using the pmi-weighted conventional features for vectorization (*Conv.* in the legend), Figure 7 using MAX_SBE (*MAX_SBE* in the legend), and Figure 8 using pmi-weighted substitution words (*Subst.* in the legend). The clustering methods are also indicated in each legend: TCRS using percentile-based similarity thresholds (*TCRS percentile*), TCRS using fixed similarity thresholds (*TCRS fixed*), k-means with k decided by the GAP statistic (*k-means gap*), k-means with fixed $k = 10$ (*k-means* $k = 10$), k-means with $k = 20$ (*k-means* $k = 20$).

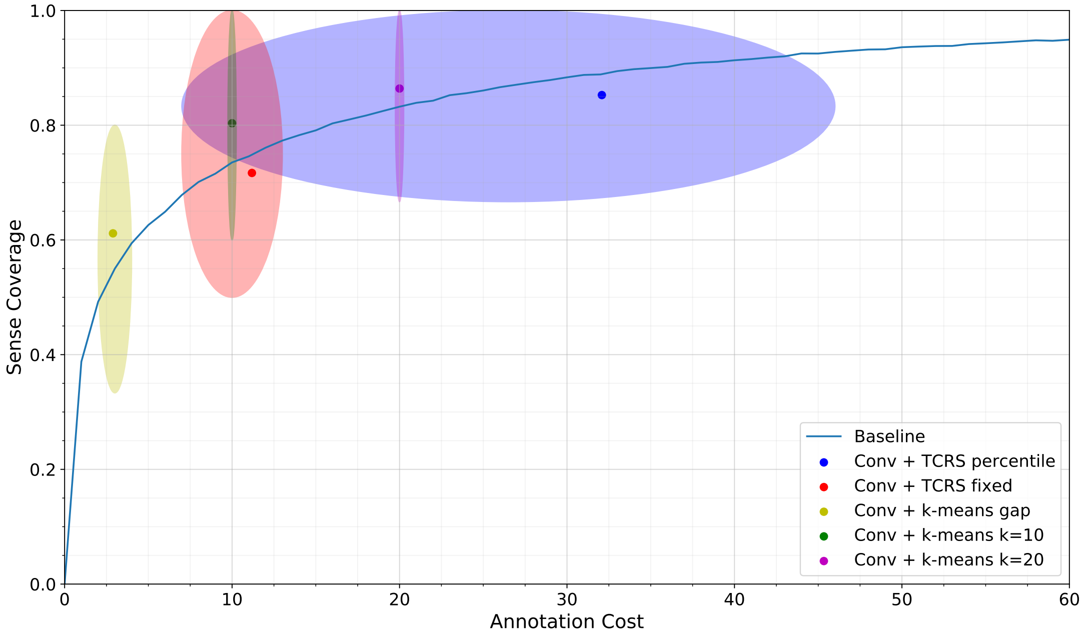

FIGURE 6: A box plot of annotation cost plotted versus sense coverage for the clustering methods that used pmi-weighted conventional features. See Section 5.1.1 and Figure 5 for further explanation on this plot.

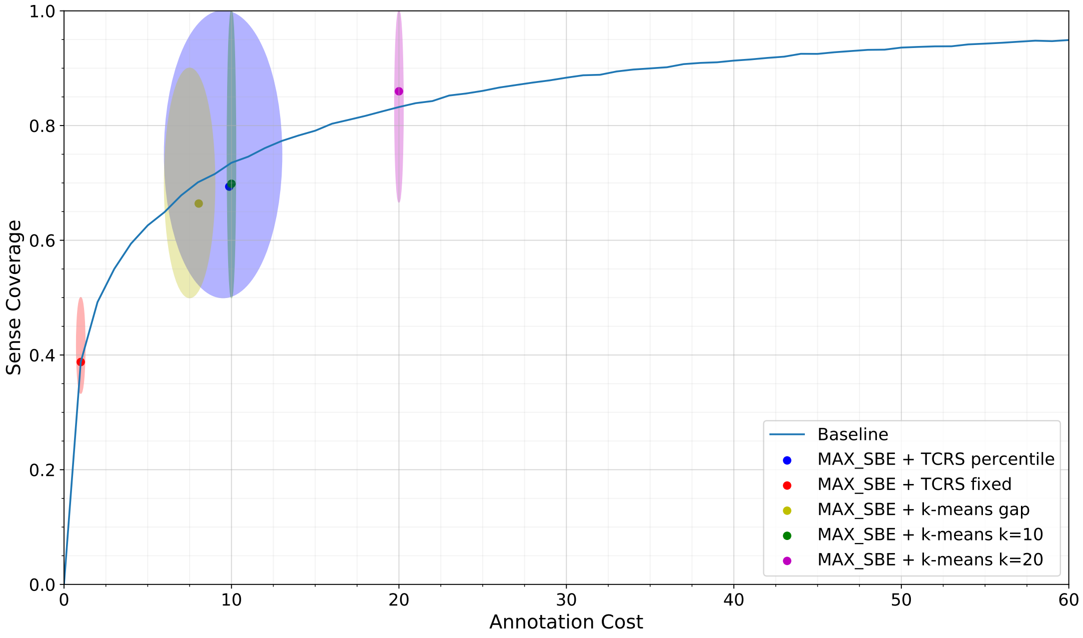

FIGURE 7: The annotation cost plotted against sense coverage for the clustering methods that used MAX_SBE. See Section 5.1.1 and Figure 5 for further explanation on this plot.

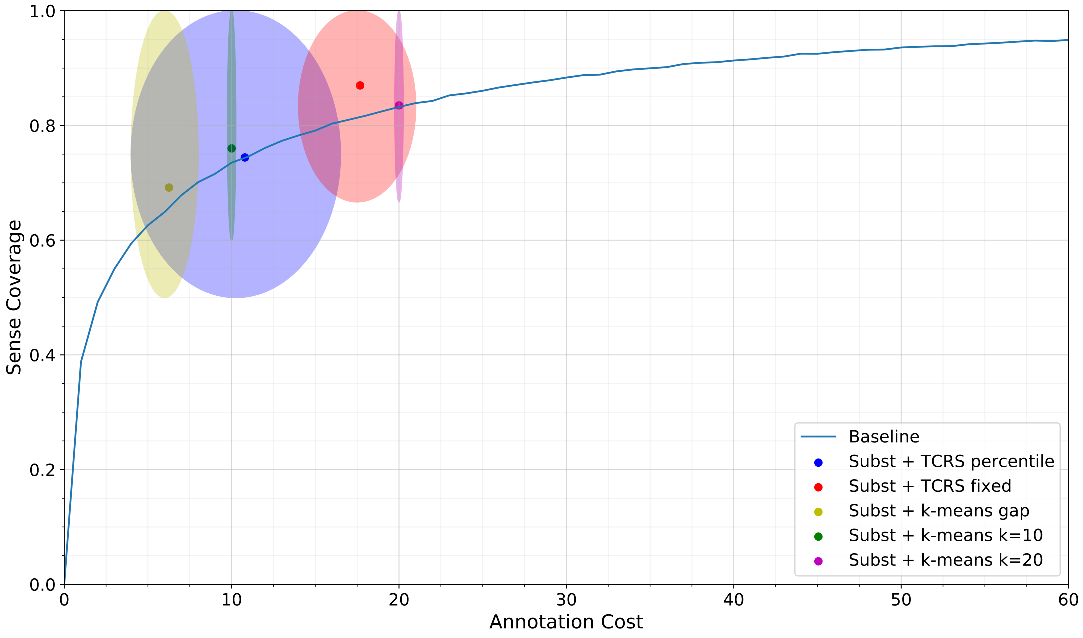

FIGURE 8: The annotation cost plotted against sense coverage for the clustering methods that used pmi-weighted substitute words. See Section 5.1.1 and Figure 5 for further explanation on this plot.

Figure 9 shows the completeness versus homogeneity for each WSI method. The WSI methods often had low completeness due to senses being spread out over many clusters. The WSI methods with the highest V-measure and homogeneity all use TCRS to clustering vectorized occurrences. TCRS could consider occurrences as noise while k-means had to cluster all occurrences, which allowed TCRS to obtain a much higher homogeneity. Note that there were only few annotated occurrences per abbreviation, which makes the validity of homogeneity and completeness debatable. Therefore, these metrics should be regarded by the reader as additional insight into the difference between WSI methods, rather than indicative of the performance of each WSI method.

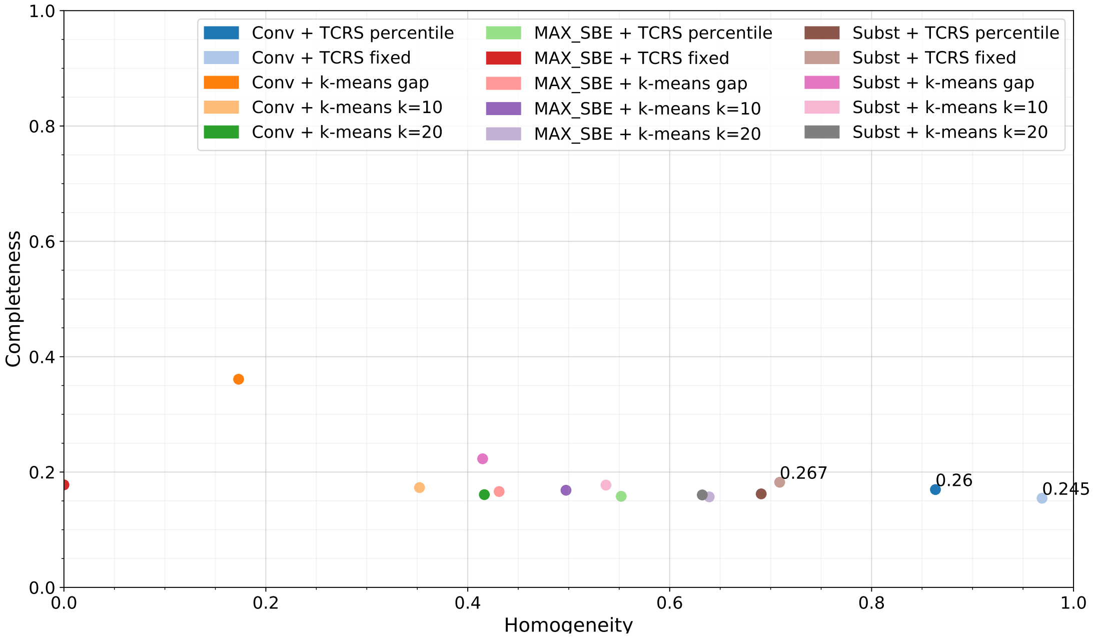

FIGURE 9: The homogeneity plotted against completeness for all WSI approaches. The three highest V-measures were annotated in the plot to indicate the most suitable WSI methods for candidate sense ranking: *Subst. + TCRS fixed* (0.267), *Conv. + TCRS percentile* (0.260), *Subst. + TCRS fixed* (0.245).

It is possible to roughly derive the annotation cost reduction and sense coverage gain for each WSI method from Figure 6, 7, and 8 (this is shown in Figure 5). To aid the reader, Table 9 shows the annotation cost reduction, sense coverage gain, mean annotation cost, mean sense coverage, and V-measure numerically rather than graphically for each WSI method.

The results show that some WSI methods can reduce annotation cost with respect to randomly annotating occurrences. In particular, I consider *Subst. + TCRS fixed* to be the best performing method. This method obtains a high mean sense coverage of 87.0% for a mean annotation cost of 17.7, which results in an annotation cost reduction of 34.5% and a sense coverage of 6.5%. Unexpectedly, conventional features clustered using k-means results in comparable annotation cost reduction and sense coverage gain, while TCRS performed poorly using conventional features. This opposes the results of Xu et al. [45], whose WSI experiment on English clinical abbreviations showed that conventional features clustered using TCRS worked better than clustering those features using

WSI method	Sense coverage	Annotation cost	Sense coverage gain	Annotation cost reduction	V-measure
Conv. + TCRS percentile	85.3% ± 21.6%	32.1 ± 23.2	-4.0%	-33.7%	0.260 ± 0.160
Conv. + TCRS fixed	71.7% ± 23.9%	11.2 ± 5.5	-3.9%	-11.9%	0.245 ± 0.145
Conv. + k-means Gap	61.1% ± 25.3%	2.9 ± 1.5	11.2%	42.2%	0.156 ± 0.210
Conv. + k-means k=10	80.3% ± 22.2%	10.0 ± 0.0	9.3%	41.2%	0.214 ± 0.183
Conv. + k-means k=20	86.4% ± 20.4%	20.0 ± 0.0	3.8%	23.1%	0.213 ± 0.178
MAX_SBE + TCRS percentile	69.4% ± 22.7%	9.9 ± 4.3	-5.7%	-23.3%	0.196 ± 0.144
MAX_SBE + TCRS fixed	38.8% ± 11.3%	1.0 ± 0.0	0.0%	0.0%	0.000 ± 0.000
MAX_SBE + k-means Gap	66.4% ± 21.9%	8.0 ± 2.3	-5.3%	-15.0%	0.221 ± 0.167
MAX_SBE + k-means k=10	69.9% ± 22.6%	10.0 ± 0.0	-5.0%	-25.0%	0.234 ± 0.162
MAX_SBE + k-means k=20	86.0% ± 19.1%	20.0 ± 0.0	3.3%	20.0%	0.235 ± 0.148
Subst. + TCRS percentile	74.4% ± 23.9%	10.7 ± 7.3	-0.2%	1.9%	0.235 ± 0.153
Subst. + TCRS fixed	87.0% ± 18.9%	17.7 ± 4.7	6.5%	34.5%	0.267 ± 0.172
Subst. + k-means Gap	69.2% ± 23.6%	6.3 ± 2.7	6.6%	21.8%	0.237 ± 0.182
Subst. + k-means k=10	76.0% ± 23.6%	10.0 ± 0.0	3.4%	16.7%	0.245 ± 0.180
Subst. + k-means k=20	83.5% ± 21.4%	20.0 ± 0.0	3.2%	4.8%	0.238 ± 0.168

TABLE 9: The numerically presented results of the WSI experiment. The metrics are given as mean and standard deviation across all abbreviations for 10 iterations of the WSI experiment.

expectation-maximization clustering, and both those clustering methods resulted in better performance than randomly selecting occurrences to annotate. The error analysis of the method for approximating sense coverage in Section 6.2.3 hints that this result might be caused by inaccuracies in the sense coverage approximation.

In contrast to the performant WSI methods, the usage of MAX_SBE vectorization results in insignificant differences from the baseline. This can have various reasons, for example, taking the maximum per word embedding dimension might not work as well for WSI as it did for WSD in other studies. It could be better to use a different embedding aggregation (e.g. mean or sum over the word embeddings) or to use the occurrence sentence instead of a context window for the word embeddings. The WSI methods using substitution lemmas showed good performance, since the means for sense coverage and annotation cost of all clustering methods were located above the baseline. Therefore, it is unlikely that MedRoBERTa.nl was to blame for MAX_SBE vectorization resulting in bad performance.

Table 10 shows the result of candidate sense ranking the annotated occurrences. Table 11 shows the result of candidate sense ranking using the clusters obtained from TCRS clustering with pmi-weighted conventional features. On average, there are 3752 candidate senses per abbreviation, and 11 out of the 45 senses are not present among the candidate senses due to not occurring 10 or more times in the corpus.

The semantic similarity ranking shows improvements in all evaluation metrics relative to the baseline (random ranking). Relaxed WMD doubles the MRR with respect to sentence similarity ranking, but comes at a much greater computational cost. Relaxed WMD took 600-1000 milliseconds to calculate for a single candidate-sense-abbreviation pair, while sentence similarity ranking took 5 milliseconds. Sentence similarity ranking only required a single distance computation for each pair of sentences, while relaxed WMD

Ranking Method	MRR	Rank	Accuracy
Random (baseline)	0.00072	2021 ± 1270	0.441
Relaxed WMD	0.02365	1272 ± 1557	0.765
Sentence similarity	0.01091	1320 ± 1533	0.735

TABLE 10: The results for ranking candidate senses based on occurrences from the ground truth dataset. The *Rank* displays the average rank and standard deviation. The *Accuracy* displays whether a sense was correctly ranked above the other senses of its abbreviation.

Ranking Method	MRR	Rank
Random (baseline)	0.00119	1165 ± 728
Relaxed WMD	0.02369	1111 ± 1403
Sentence similarity	0.01106	986 ± 1123

TABLE 11: The results for ranking candidate senses based on occurrences from clusters made with TCRS and pmi-weighted conventional features. The *Rank* displays the average rank and standard deviation.

required taking the row-wise minimum of $N \times M$ distance computations, where N was the number of words in one sentence, and M in the other sentence.

Nevertheless, the mean ranks of semantic similarity ranking are too high to be useful in a practical setting. Only 3 senses ranked in the top-20, and none at the top of a ranking. This result indicates that candidate sense ranking is not useful for automatic sense retrieval, or as additional information to expert annotators. However, an analysis of the candidate senses in the top-20 of semantic similarity rankings showed that these rankings are not so bad after all. This analysis is included in the error analysis of candidate sense ranking in Section 6.2.2.

6.2 Error Analysis

Subst. + TCRS fixed obtains high sense coverage, annotation cost reduction and V-measure, which makes it most suitable to build a sense inventory. Therefore, the error analysis is focused on this WSI method. The high variance in sense coverage warrants an analysis of how well each abbreviation and its senses are clustered, which is shown in Section 6.2.1). Similarly, I analyze the position of each abbreviation's sense in the candidate sense ranking in Section 6.2.2.

The evaluation metric *sense coverage* is approximated based on a limited number of annotations. I analyze in Section 6.2.3 how accurate the approximated sense coverage is by labelling the centroids of each abbreviation and each WSI method. This is not an error analysis on the methods used in this experiment, but rather an error analysis of an evaluation metric, which is not standard procedure. However, the approximated sense coverage error analysis allows for further contextualization of the results of the WSI experiment, so it provides valuable insight to this thesis.

6.2.1 Word-Sense Induction

WSI can make three types of errors: 1) considering all annotated occurrences of a sense as noise, 2) putting a sense in an inhomogeneous cluster, and 3) spreading a sense over many clusters. The first error leads to lower sense coverage, since those senses do not occur in any cluster. The second error also leads to lower sense coverage, since an inhomogeneous cluster contains occurrences of different senses, and only the centroid represents the cluster. This error is quantified as the precision of the best cluster for a sense. The third error leads to higher annotation cost, since each cluster representing the same sense leads to an additional annotation without increasing sense coverage. This error is quantified as the recall of the best cluster for a sense. The best cluster for a sense is the one resulting in the highest F1-score. The quantification of these errors per sense is visible in Appendix C.

Firstly, the abbreviation senses are categorized based on their sense frequency, which is shown in Table 12. The frequency, noise percentage, precision, and recall are averaged over all senses in a category to quantify the prevalence of each error type per sense category. It becomes clear that occurrences of rare senses (0%-5%) are more prone to error type 1, since there are more senses that only have noise occurrences. This is not only due to the rare senses having less occurrences, since the average noise percentage should then still be the similar to that of other frequency categories. The likely reason is that occurrences of rare senses have no other occurrences that are similar enough to be clustered together, causing all of them to be considered noise. This observation indicates that the best WSI method is not suitable in creating sense clusters for rare senses with too much variation among its occurrence vectors, i.e. WSI does not work for rare senses that occur in wildly dissimilar contexts.

Frequency category	**Avg. frequency**	**Only noise**	**Avg. noise%**	**Avg. precision**	**Avg. recall**
50% - 100%	83.4%	0 / 13	36.8%	94.3%	27.5%
10% - 50%	24.4%	1 / 14	27.7%	91.3%	43.9%
5% - 10%	7.4%	1 / 6	39.4%	62.3%	23.8%
0% - 5%	2.9%	5 / 10	60.0%	27.9%	35.0%
0% - 100%	34.8%	7 / 43	39.6%	73.4%	34.0%

TABLE 12: Quantification of errors for senses categorized by sense frequency. *Avg. Frequency* is the average frequency of senses in the category. *Only Noise* is the fraction of the senses for which all annotated occurrences are considered as noise. *Avg. Noise%* is the percentage of occurrences that are noise averaged over the senses in the category. *Avg. Precision* and *Avg. Recall* are the precision and recall of the best cluster averaged over the senses in the category.

The average precision is contextualized by the average frequency per frequency category. The average precision of very frequent (50% - 100%) senses is 94.3%, but putting all occurrences in a single cluster would already lead to an average precision of 83.4%, which is the average frequency. The average precision of infrequent (5% - 10%) and rare (0% - 5%) senses appears quite low, but is quite high relative to the average frequency. Sense frequency appears to be correlated with precision, which leads to extremely inhomogeneous clusters for rare senses (error type 2). These impure clusters indicate inadequate merging of tight clusters, or inadequate vectorization such that two occurrences of different senses lie close in vector space. The merging or vectorization are inadequate in the context of clustering senses.

The average recall is quite low for all frequency categories, even when considering the percentage of occurrences that are considered noise. WSI also results in 17.7 clusters on average per abbreviation, which is mostly due to frequent senses, since they simply have more occurrences. This means that frequent senses are the largest contributor to error type 3. Again, the problem lies with merging or vectorization. A solution would be to lower the similarity threshold for merging, but this could come at the cost of lowering cluster precision and therefore sense coverage. Another solution would be to only annotate centroids of the largest clusters, which also could lower sense coverage, but can greatly reduce annotation cost.

Secondly, the abbreviation senses are categorized by their healthcare domain, which is shown in Table 13. Though the average frequency does not differ much between domains, Table 14 shows that the distribution of frequency category per healthcare domain is very different for mental healthcare. It becomes clear that the mental healthcare sector has many more rare and infrequent senses relative to the elderly care and disability care sectors. This large difference could be (in part) caused by the small size of the dataset, rather than being distinctive for mental healthcare. As a result, these senses are clustered more poorly: the average noise percentage is higher, and average precision and recall are lower.

Healthcare domain	**Avg. frequency**	**Only noise**	**Avg. noise%**	**Avg. precision**	**Avg. recall**
Elderly care	35.7%	1 / 14	26.6%	86.5%	41.6%
Mental healthcare	33.3%	4 / 12	52.3%	52.1%	31.8%
Disability care	35.3%	2 / 17	41.3%	77.9%	35.1%
All	34.8%	7 / 43	39.6%	73.4%	34.0%

TABLE 13: Quantification of errors for senses categorized by healthcare domain. See the caption of Table 12 for an explanation on each column.

Frequency category → **Domain ↓**	0% - 5%	5% - 10%	10% - 50%	50% - 100%
Elderly care	14.3%	21.4%	35.7%	28.6%
Mental healthcare	41.7%	16.7%	8.3%	33.3%
Disability care	17.6%	5.9%	47.1%	29.4%

TABLE 14: The overlap between sense frequency and healthcare domain. The percentages are relative to the healthcare domain. For example, 14.3% of the elderly care abbreviation senses have a frequency between 0% and 5%.

Despite the overlap in sense frequency and healthcare domain, the elderly care sector appears to be clustered better than the other domains. A reason could that the healthcare reports from this sector are more often related to the medical domain than the reports from the other two sectors. MedRoBERTa.nl, which is used in the best WSI method, is trained specifically on clinical reports. Therefore, the WSI method performs less well on texts outside the medical domain. Elderly care reports are largely made for nursing visits, while mental healthcare reports are made for therapy and counseling sessions, and disability care reports are made for counseling sessions. This is even visible in the dataset, since the abbreviations "*ab*" and "*hb*" are used in both elderly and disability care. The medical senses of each abbreviation have a frequency greater than 90% in elderly care reports, while they have a frequency less than 15% in disability care reports. Similarly, the mental healthcare domain has no medical senses.

A summary of the error analysis of the best WSI method:

- Occurrences of rare senses are more often considered noise due to absence of occurrences of the same sense with similar context;
- Infrequent and rare senses are often put in inhomogeneous clusters (i.e. clusters with low precision);
- Very frequent senses are the main cause of unnecessary annotation cost;
- The elderly care senses are clustered better, likely because the reports from that domain are more closely related to medical reports than reports from mental healthcare and disability care. Medical reports were used to train the MedRoBERTa.nl model, which is used in the best WSI method.

6.2.2 Candidate Sense Ranking

Candidate sense ranking using semantic similarity measures resulted in varying performance compared to the baseline. 3 out of the 34 senses ranked in the top 20 using WMD, which has a probability of roughly 0.9% of occurring through random ranking. Meanwhile, 15 out of the 34 senses had a rank over 1000, occasionally worse than the baseline rank. The exact ranks per sense can be seen in Appendix C. I concluded that candidate sense ranking is not useful for automatic sense retrieval or annotation assistance, since there are so few senses that rank well. Therefore, I reflect only briefly on the relation between ranking performance and sense frequency or healthcare domain.

Table 15 shows the performance of each ranking method per frequency category. I hypothesized that comparing more abbreviation occurrences to more occurrences of candidate senses would lead to a more accurate ranking, since many occurrences are more representative of the semantics of a sense than a few. This hypothesis is not confirmed in Table 15, because the semantic similarity ranking for rare and infrequent senses rank is similar to that of frequent and very frequent senses relative to the baseline ranking.

Table 15 shows the performance of each ranking method per healthcare domain. Relative to the baseline, the semantic similarity ranking for elderly care and mental healthcare performs similar, but it performs worse for disability care.

Additionally, I looked into the top-20 ranked candidate senses per abbreviation sense. I encountered completely irrelevant candidate senses near the top of the ranking, such as "*houtwerkplaats bang (wood workshop scared)*" at rank 4 for the sense "*hb = hemaglobine (haemoglobin)*". On the other hand, I also encountered candidate senses related

Frequency category	**Present as candidate sense**	**Relaxed WMD rank**	**Sentence similarity rank**	**Baseline rank**
50% - 100%	13 / 13	1045 ± 1142	1117 ± 1088	1612 ± 985
10% - 50%	10 / 14	1734 ± 2133	1641 ± 2080	1969 ± 1064
5% - 10%	4 / 6	874 ± 893	854 ± 762	2282 ± 1347
0% - 5%	7 / 10	1259 ± 1377	1503 ± 1524	2702 ± 1602
0% - 100%	34 / 43	1272 ± 1557	1320 ± 1533	2021 ± 1270

TABLE 15: Results of candidate sense ranking categorized by sense frequency ranges.

Healthcare domain	**Present as candidate sense**	**Relaxed WMD rank**	**Sentence similarity rank**	**Baseline rank**
Elderly care	10 / 14	846 ± 1320	811 ± 1177	1462 ± 726
Mental healthcare	12 / 12	1384 ± 1077	1465 ± 1215	2530 ± 1357
Disability care	12 / 17	1514 ± 2007	1598 ± 1928	1977 ± 1330
All	34 / 43	1272 ± 1557	1320 ± 1533	2021 ± 1270

TABLE 16: Results of candidate sense ranking categorized by healthcare domain.

to the abbreviation senses, such as "*huisbezoek voor (house visitation for)*" at rank 8 for the sense "*hb = huisbezoek (house visitation)*", while the exact sense had rank 2237. This type of false negative could indicate that candidate sense ranking has more potential than appears from Table 10 and 11.

The presence of candidate senses related to an abbreviation sense is hard to quantify, since my judgement of n-gram similarity is subjective. Therefore, I denote a top-20 as informative, if it contains a completely unabbreviated synonym of the abbreviation sense. For example, the candidate sense "*antibiotica kuur (antibiotics treatment)*" is informative enough to derive the sense "*antibiotica (antibiotics)*", but the candidate sense "*huisb (house v)*" is not informative enough to derive "*huisbezoek (house visitation)*". Similarly, the candidate sense "*haar bloedsuiker (her blood sugar)*" is similar, but not informative enough to derive "*hemaglobine (haemoglobin)*".

The quantified inspection of top-20 candidate sense ranking inspection is shown in Table 17. It becomes clear that candidate sense ranking is much better if the objective is not to find the exact sense. This could mean that showing the top-20 ranked candidate senses to an annotator allows them to annotate abbreviations that they are stuck on. Furthermore, the highest ranking sense is informative of the abbreviation sense 30% of the time, so showing that to the annotator could also help.

I commonly observed high-ranking candidate senses that contained the exact sense or synonym together with a preposition, conjunction or verb. For example, the sense "*huisbezoek (house visitation)*" has rank 2237 and 2001 using relaxed WMD and sentence similarity respectively, but the top-20s of these two rankings contain the exact sense followed by "*te* (to [verb] / too)", "*om (at [time])*", '*voor (before / for)*', and '*kan (can = verb)*'. Finding the exact reason for why this happens requires an entire new study, which is outside the scope of this thesis. Nevertheless, this error analysis shows that se-

Ranking method	**Top-20 contains informative candidate**	**Rank**	**Ranked at top**
Relaxed WMD	25 / 34	6 ± 6	9 / 34
Sentence similarity	21 / 34	3 ± 3	10 / 34

TABLE 17: Results of inspecting the top-20 ranked candidate senses. *Top-20 contains informative candidate* shows how many of the senses had a candidate in the top-20 that is an inflection or synonym of the sense. *Rank* indicates the average rank of the informative candidate sense and the standard deviation in rank. *Ranked at top* indicates how many times the informative candidate sense had the top rank.

mantic similarity ranking has great potential for further automating abbreviation sense retrieval or lexical substitute retrieval. This potential for future work is further addressed in Section 7.2.

6.2.3 Sense Coverage Approximation

The sense coverage was approximated for each abbreviation clustering using the limited number of annotated occurrences provided in the ground truth dataset (as shown in Figure 4). To validate how accurate this approximation was, I (a non-expert) have annotated the centroids of each WSI method's abbreviation clustering for a single experiment run, allowing me to calculate the exact sense coverage for that experiment. There are 15 WSI methods, 15 abbreviations, and 13.5 centroids per clustering on average, so I annotated a little over 3,000 centroids in total. This was a closed-label annotation task, while the annotation task for the ground truth was open-label, which allowed for the larger number of annotations.

The approximated and exact sense coverages are shown per WSI method in Table 18. The goal of this error analysis was to discover whether the approximated sense coverage inflates or deflates the exact sense coverage for any of the WSI methods. For example, if *Subst. + TCRS fixed* had a much lower exact sense coverage than the approximated sense coverage, it could indicate that this is actually not the best WSI method to use for building a sense inventory for Dutch healthcare abbreviations. In contrast, if the approximated sense coverage is close to the exact sense coverage, it strengthens the results discussed in Section 6.1. Table 18 shows that the exact sense coverage is on average 0.5% lower than the approximated sense coverage for the same experiment, and is 1% lower than the approximated sense coverage over 10 experiments. Most notably, *Conv. + k-means k=20* has an exact sense coverage that is 12.2% lower than the approximated sense coverage for the same experiment, and 11.0% lower than the approximated sense coverage over 10 experiments. *Subst. + TCRS fixed*, which I consider to be the best WSI method among those tested, has an exact sense coverage that is 1.3% higher than the approximated sense coverage for the same experiment, and the same as the approximated sense coverage over 10 experiments.

From the sense coverages shown in Table 18, it does not appear that the approximated sense coverage systematically inflates or deflates the exact sense coverage over all WSI methods.

WSI method	**Sense coverage (1)**	**Sense coverage (2)**	**Sense coverage (3)**
Conv. + TCRS percentile	85.4%	87.6%	85.3%
Conv. + TCRS fixed	75.8%	70.2%	71.7%
Conv. + k-means Gap	58.1%	58.6%	61.1%
Conv. + k-means k=10	75.6%	82.9%	80.3%
Conv. + k-means k=20	76.9%	87.6%	86.4%
MAX_SBE + TCRS percentile	72.1%	68.8%	69.4%
MAX_SBE + k-means Gap	71.3%	68.2%	66.4%
MAX_SBE + k-means k=10	69.7%	66.9%	69.9%
MAX_SBE + k-means k=20	79.6%	81.4%	86.0%
Subst. + TCRS percentile	78.0%	72.4%	74.4%
Subst. + TCRS fixed	87.0%	85.9%	87.0%
Subst. + k-means Gap	68.6%	74.1%	69.2%
Subst. + k-means k=10	75.2%	74.3%	76.0%
Subst. + k-means k=20	83.1%	83.3%	83.5%
All	75.5%	75.8%	76.2%

TABLE 18: The average sense coverage per abbreviation for each WSI method (except *Subst + TCRS fixed*) calculated in three different ways: 1) the exact sense coverage as per my centroid annotations, 2) the approximated sense coverage for a single experiment (same experiment as (1)), and 3) the approximated sense coverage averaged over 10 experiments (also shown in Table 9). At the bottom, the average sense coverage over all methods is shown.

The annotation of centroids allowed for additional interesting discoveries. These discoveries are summarized in the list below:

- Centroids contained non-abbreviations, such as postal codes (fictional example: "*[ID] AB Amsterdam*") where the digits of the postal code were deidentified as an ID number), names (e.g. "*Ab*" is a Dutch name), and initials (fictional example: "*G.G. Modaal*") that were missed by the deidentification tool. These non-abbreviation centroids don't impede the sense coverage, but increase annotation cost, since additional centroids are annotated that do not contain an abbreviation sense. Their prevalence was roughly 1 in every 15 centroids across all WSI methods.

- The sense "*hoofd hals*" for the elderly care abbreviation "*hh*" never occurred among 226 (not necessarily unique) centroids. A possibility is that all WSI methods fail to cluster this sense adequately. However, it could be that the sense occurred among the 60 annotated occurrences by chance, and actually has a sense frequency far lower than 1.8%, since the sense only had a single occurrence. Similarly to "*hoofd hals*", the senses "*gezins*" and "*gegeven*" also never occurred among the centroids. If these sense frequencies are indeed inflated, the baseline can gain an unfair advantage over the WSI methods, since it samples such senses based on their inflated frequency.

- New senses were discovered, which are shown in Table 19. Table 19 is similar to the table in Section 4, but the number of observations (annotated centroids) is given rather than the sense frequency. Note that roughly 200 centroids were annotated per abbreviation on average.

Domain	**Abbr-eviation**	**Sense**	**Translated sense**	**#Observations**
elderly care	wv	wondverpleegkundige	woundcare specialist	1
	ab	absorberend	absorbing	3
mental healthcare	bw	begeleid wonen	assisted living	unsure[1]
	gz	groepszorg	group care	>50
disability care	ib	inkomensbelasting[2]	income tax	1
	ab	afstandsbediening	remote control	1

TABLE 19: List of senses that I discovered when annotating WSI centroids as a non-expert.

[1] The sense "*assisted living*" was rarely distinguishable from "*protected living*" during centroid annotation, since I am a non-expert on mental healthcare.

[2] This sense origins from the IB60 form, which is now called the "*inkomensverklaring*" (= "*income statement*").

7 Discussion

The results show that the best WSI method, *Subst. + TCRS fixed*, obtains a sense coverage of 87.0% on average and reduces annotation cost with 34.5% compared to randomly annotating occurrences to obtain the same sense coverage. Therefore, this WSI method can be used to build a sense inventory for Dutch healthcare abbreviations efficiently. Such a sense inventory can then be used by any reader of healthcare reports to look for senses that they are not familiar with. Furthermore, this sense inventory can be used for future NLP applications and research using Dutch healthcare reports, such as query abbreviation expansion for information retrieval, or disambiguating abbreviations. If all clusters are assigned a sense through the proposed WSI method, these clusters could even be used for abbreviation disambiguation, for example through nearest neighbor classification. This has been done for English clinical abbreviations by Wu et al. [42], who developed a pipeline called Clinical Acronym Recognition and Disambiguation, or CARD for short.

WSI for building a sense inventory mainly reduces annotation cost for abbreviations with multiple senses. It might be that Dutch healthcare abbreviations often have a single sense, similar to 62.7% of English clinical abbreviations in the sense inventory by Moon et al. [26]. It mainly depends on the use-case whether WSI should be used for abbreviation sense retrieval. If the goal is to capture is many abbreviation senses as possible for the smallest number of annotations, a single annotation per abbreviation is most efficient, but the sense inventory would likely be far from complete. For precise abbreviation disambiguation or query abbreviation expansion, a more complete sense inventory should be built, which is when WSI is useful. Additionally, the sense inventory by Moon et al. showed that the longer an abbreviation is, the fewer senses it likely has. Perhaps WSI can be used for retrieving senses of abbreviations shorter than 5 letters, since a longer abbreviation is likely to be an acronym with a single sense.

The WSI error analysis in Section 6.2.1 ends in a summary of four observations. The first and second observation indicate that clustering rare senses (frequency below 5%) is most prone to errors. This can be mitigated by increasing the amount of data used for WSI and making the similarity bounds for TCRS more strict. There is far more data available to Nedap, and the main challenge would be to not let the number of clusters increase too much. This relates to the third observation: frequent senses are spread over many clusters. A solution to this problem would be to devise an annotation strategy for the clusters resulting from WSI. Instead of annotating all cluster centroids, experts could only annotate the centroids of the largest clusters, or the centroids of clusters that are most distant from each other. The fourth and last observation entails that elderly care abbreviations clustered better than those from mental healthcare or disability care. A solution to this problem would be to further pre-train MedRoBERTa.nl on each healthcare domain. Another option would be to train a RoBERTa model from scratch for each healthcare domain, but this would be more computationally intensive.

Candidate sense ranking using semantic similarity measures showed some valuable results during error analysis, but the method is not viable for immediate practical application. Future work should focus on post-processing the highest ranking candidate senses, since those often contain the exact sense or a synonym. Furthermore, semantic similarity measures can be incorporated in TCRS clustering for WSI instead of cosine similarity.

7.1 Limitations

The main limitation of this thesis lies with the ground truth dataset. A problem that this thesis tackles is that healthcare professionals do not have the time to annotate many occurrences of the same abbreviation for creating a sense inventory. Ironically, annotating many occurrences is required for making a large ground truth dataset, such that WSI methods to reduce the number of annotations can be evaluated accurately. The dataset in this thesis only contained 15 abbreviations. Additionally, only a few annotations were made per abbreviation, which was further reduced by annotators being unable to annotate 10 out of 60 occurrences on average. Therefore, I had to be cautious with making strong conclusions based on the results and conduct an extensive error analysis.

I mitigated the smallness of the dataset by running the WSI experiment 10 times, and by using the ground truth sense frequencies to estimate the sense of clusters without annotated occurrences. An error analysis on the approximated sense coverage showed that, on average, it is close to the exact sense coverage, which shows that my mitigation strategy was somewhat effective.

Another limitation resides with the configuration of the TCRS similarity thresholds. These thresholds were either estimated based on occurrence similarity percentiles, or adopted from Xu et al. [45], who optimized them for their dataset. These were not optimized in this thesis due to it further reducing the already limited ground truth dataset. Now that TCRS clustering resulted in the best WSI method, it would be interesting to explore how annotation cost and sense coverage can be improved by optimizing the similarity thresholds on these measures.

7.2 Future Work

Plenty of future work is possible based on the outcome of this thesis. The WSI error analysis and my discussion above shows various suggestions to improve WSI, but these are relatively minor improvements. I mainly suggest that future studies focus on making candidate sense ranking practically useful for abbreviation sense retrieval. The challenge will entail identifying candidate senses that contain inflections of a word or group of words, and grouping those together. Such an approach should be accommodated with a practical evaluation measure. Not only the rank of a group of candidate senses should be measured, but also the time spend on an annotation by an expert. A research question could be: "*to what extent can abbreviation occurrence annotation be sped up by showing the top-5 ranked candidate senses?*"

Furthermore, future work could investigate whether a strategic order can be devised for annotating cluster centroids obtained through WSI. This could be beneficial to abbreviation sense retrieval tooling, since end-users can define how many annotations they want to make per abbreviation. The strategy should mainly take advantage of the WSI error that very frequent senses are spread across many clusters. For example, the centroids could be annotated in order of their cluster size, or on how far the centroid is positioned from other centroids.

8 Conclusion

In Section 1.1, I formulate a research question and split it up into three sub-questions: sub-RQ1 regards using WSI to reduce annotation cost, sub-RQ2 regards using WSI to gain sense coverage, and sub-RQ3 regards using semantic similarity measures to accurately rank candidate senses of an abbreviation.

I address sub-RQ1 and sub-RQ2 through experimenting with various WSI methods and evaluating them with respect to a baseline of random annotation. Various WSI methods showed to outperform the baseline, and the best WSI method resulted in an annotation cost reduction of 34.5% with an average sense coverage of 87.0%, equivalent to a gain in sense coverage of 6.5% with an average annotation cost of 17.7. The method entailed using MedRoBERTa.nl to predict substitutes in place of an abbreviation, which were then lemmatized, pmi-weighted and clustered using TCRS clustering. This result shows that WSI can aid in reducing the annotation effort and increasing sense coverage for building a sense inventory, thus showing positive results for sub-RQ1 and sub-RQ2. If this method were to be further optimized and more extensively evaluated, perhaps even better results could be obtained.

I address sub-RQ3 by comparing two different semantic similarity measures for ranking candidate senses with respect to a random ranking baseline. The rankings of exact senses are poor, but high ranking candidate senses often include the exact sense, a synonym, or inflection. Future work is necessary to further evaluate the usefulness of semantic similarity measures and candidate sense ranking in building a sense inventory. Relaxed WMD for candidate sense ranking shows better results than sentence similarity, but is drastically more computationally expensive.

To conclude, the main research question is addressed adequately through the conclusions on the three formulated sub-questions. The best WSI method can be applied directly for building a sense inventory for abbreviations from Dutch healthcare reports more efficiently. Meanwhile, the candidate sense ranking was not adequate to directly be integrated in building a sense inventory, yet it holds the potential that it might be used to automatically retrieve senses or aid annotators in the future.

References

[1] D. Alagić, J. Šnajder, and S. Padó. Leveraging Lexical Substitutes for Unsupervised Word Sense Induction. In *AAAI 2018*, pages 5004–5011, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/12017.

[2] A. Amrami and Y. Goldberg. Word Sense Induction with Neural biLM and Symmetric Patterns. In *EMNLP 2018*, pages 4860–4867. Association for Computational Linguistics, 2018. ISBN 9781948087841. doi: 10.18653/v1/d18-1523.

[3] H. Ao and T. Takagi. ALICE: An Algorithm to Extract Abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 12(5):576–586, sep 2005. ISSN 10675027. doi: 10.1197/JAMIA.M1757.

[4] S. Brody and M. Lapata. Bayesian Word Sense Induction. In *EACL 2009*, page 30. Association for Computational Linguistics, 2009. URL https://aclanthology.org/E09-1013.pdf.

[5] R. Chasin, A. Rumshisky, O. Uzuner, and P. Szolovits. Word sense disambiguation in the clinical domain: a comparison of knowledge-rich and knowledge-poor unsupervised methods. *JAMIA*, 21(5):842–849, 2014. ISSN 1527-974X. doi: 10.1136/AMIAJNL-2013-002133.

[6] D. Chopard and I. Spasić. A Deep Learning Approach to Self-expansion of Abbreviations Based on Morphology and Context Distance. In *Statistical Language and Speech Processing*, volume 11816, pages 71–82. Springer International Publishing, 2019. ISBN 9783030313715. doi: 10.1007/978-3-030-31372-2_6.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, mar 2011. URL http://arxiv.org/abs/1103.0398.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North*, pages 4171–4186, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[9] A. Di Marco and R. Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013. ISSN 15309312. doi: 10.1162/COLI_A_00148.

[10] M. J. Dooley, M. Wiseman, and G. Gu. Prevalence of error-prone abbreviations used in medication prescribing for hospitalised patients: multi-hospital evaluation. *Internal Medicine Journal*, 42(3):e19–e22, mar 2012. ISSN 14440903. doi: 10.1111/J.1445-5994.2011.02697.X.

[11] G. P. Finley, S. V. Pakhomov, R. McEwan, and G. B. Melton. Towards Comprehensive Clinical Abbreviation Disambiguation Using Machine-Labeled Training Data. In *AMIA Annu Symp Proc*, volume 2016, pages 560–569, 2016. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5333249.

[12] K. Goyal and E. Hovy. Unsupervised Word Sense Induction using Distributional Statistics. In *COLING 2014*, pages 1302–1310, Dublin, Ireland, 2014. URL https://aclanthology.org/C14-1123.

[13] A. M. Istaiti. Abbreviation extraction and normalization in Spanish clinical text. In *CEUR Workshop Proceedings*, volume 2633, pages 13–19, 2020. URL http://ceur-ws.org/Vol-2633/paper3.pdf.

[14] A. Jaber and P. Martínez. Disambiguating Clinical Abbreviations using Pre-trained Word Embeddings. In *BIOSTEC 2021*, volume 5, pages 501–508, 2021. ISBN 9789897584909. doi: 10.5220/0010256105010508.

[15] D. Jurgens and I. Klapaftis. SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *SemEval 2013*, volume 2, pages 290–299, Atlanta, Georgia, USA, 2013. URL https://aclanthology.org/S13-2049.

[16] I. P. Klapaftis and S. Manandhar. Evaluating word sense induction and disambiguation methods. *Language Resources and Evaluation*, 47(3):579–605, sep 2013. ISSN 1574020X. doi: 10.1007/S10579-012-9205-0.

[17] K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. F. Jones, R. Forshee, M. Walderhaug, and T. Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review. *Journal of Biomedical Informatics*, 73:14–29, sep 2017. ISSN 15320464. doi: 10.1016/J.JBI.2017.07.012.

[18] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From Word Embeddings To Document Distances. In *ICML 2015*, volume 37, pages 957–966, 2015. URL https://proceedings.mlr.press/v37/kusnerb15.pdf.

[19] C. Li, L. Ji, and J. Yan. Acronym Disambiguation Using Word Embedding. In *AAAI 15*, pages 4178–4179, 2015. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewFile/9404/9721.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, and P. G. Allen. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, jul 2019. doi: 10.48550/arxiv.1907.11692. URL https://arxiv-org.ezproxy2.utwente.nl/abs/1907.11692v1.

[21] S. Manandhar, I. P. Klapaftis, D. Dligach, and S. S. Pradhan. SemEval-2010 Task 14: Word Sense Induction & Disambiguation. In *SemEval 2010*, pages 63–68, Uppsala, Sweden, 2010. Association for Computational Linguistics. URL https://aclanthology.org/S10-1011.pdf.

[22] S. Manzar, A. K. Nair, M. Govind Pai, and S. Al-Khusaiby. Use of abbreviations in daily progress notes. *Archives of Disease in Childhood - Fetal and Neonatal Edition*, 89(4):F374, 2004. ISSN 13592998. doi: 10.1136/ADC.2003.045591.

[23] A. Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3): 289–318, 2002. ISSN 08912017. doi: 10.1162/089120102760275992.

[24] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. *Advances in neural information processing systems*, pages 3111–3119, 2013.

[25] S. Moon, S. Pakhomov, and G. B. Melton. Automated Disambiguation of Acronyms and Abbreviations in Clinical Texts: Window and Training Size Considerations. In *AMIA Annu Symp Proc*, volume 2012, pages 1310–1319, 2012. URL https://pubmed.ncbi.nlm.nih.gov/23304410/.

[26] S. Moon, S. Pakhomov, N. Liu, J. O. Ryan, and G. B. Melton. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, 21 (2):299–307, 2014. ISSN 10675027. doi: 10.1136/AMIAJNL-2012-001506.

[27] D. L. Mowery, B. R. South, L. Christensen, L.-M. Murtola, S. Salanterä, H. Suominen, D. Martinez, N. Elhadad, S. Pradhan, G. Savova, and W. W. Chapman. Task 2: ShARe/CLEF eHealth Evaluation Lab 2013. In *CLEF 2013*. CEUR Workshop Proceedings, 2013. URL http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-MoweryEt2013.pdf.

[28] R. Navigli. Word sense disambiguation. *ACM Computing Surveys*, 41(2):1–69, feb 2009. ISSN 03600300. doi: 10.1145/1459352.1459355.

[29] S. V. Pakhomov, G. Finley, R. McEwan, Y. Wang, and G. B. Melton. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, 32(23):3635–3644, 2016. ISSN 1367-4811. doi: 10.1093/BIOINFORMATICS/BTW529.

[30] A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann. Unsupervised Does Not Mean Uninterpretable: The Case for Word Sense Induction and Disambiguation. In *EACL 2017*, volume 1, pages 86–98. Association for Computational Linguistics (ACL), 2017. ISBN 9781510838604. doi: 10.18653/V1/E17-1009.

[31] N. Pérez, I. Montoya, G.-P. Aitor, and M. Cuadros. Vicomtech at BARR2: Detecting biomedical abbreviations with ML methods and dictionary-based heuristics. In *CEUR Workshop Proceedings*, volume 2150, pages 322–328, 2018. URL http://ceur-ws.org/Vol-2150/BARR2_paper5.pdf.

[32] T. Qian, D. Ji, M. Zhang, C. Teng, and C. Xia. Word Sense Induction Using Lexical Chain based Hypergraph Model. In *COLING 2014*, pages 1601–1611, Dublin, Ireland, 2014. URL https://aclanthology.org/C14-1152.

[33] J. E. Sheppard, L. C. Weidner, S. Zakai, S. Fountain-Polley, and J. Williams. Ambiguous abbreviations: an audit of abbreviations in paediatric note keeping. *Archives of Disease in Childhood*, 93(3):204–206, mar 2008. ISSN 00039888. doi: 10.1136/ADC.2007.128132.

[34] M. Skreta, A. Arbabi, J. Wang, E. Drysdale, J. Kelly, D. Singh, and M. Brudno. Automatically disambiguating medical acronyms with ontology-aware deep learning. *Nature Communications*, 12:5319, 2021. ISSN 20411723. doi: 10.1038/s41467-021-25578-4.

[35] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 63(2):411–423, 2001. doi: 10.1111/1467-9868.00293.

[36] J. Trienes, D. Trieschnigg, C. Seifert, and D. Hiemstra. Comparing Rule-based, Feature-based and Deep Neural Methods for De-identification of Dutch Medical Records. Technical report, University of Twente, 2020.

[37] K. Vanopstal, B. Desmet, and V. Hoste. Towards a learning approach for abbreviation detection and resolution. *LREC 2010*, pages 1043–1049, 2010. URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/737_Paper.pdf.

[38] S. Verkijk and P. Vossen. MedRoBERTa.nl: A Language Model for Dutch Electronic Health Records. *Computational Linguistics in the Netherlands Journal*, 11:141–159, 2021. URL https://www.clinjournal.org/clinj/article/view/132.

[39] R. V. Vidhu Bhala and S. Abirami. Trends in word sense disambiguation. *Artificial Intelligence Review*, 42(2):159–171, mar 2012. ISSN 1573-7462. doi: 10.1007/S10462-012-9331-5. URL https://link-springer-com.ezproxy2.utwente.nl/article/10.1007/s10462-012-9331-5.

[40] Y. Wu, B. Tang, M. Jiang, S. Moon, J. C. Denny, and H. Xu. Clinical acronym/abbreviation normalization using a hybrid approach. In *CLEF*, volume 1179, 2013. URL http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-WuEt2013.pdf.

[41] Y. Wu, J. Xu, Y. Zhang, and H. Xu. Clinical Abbreviation Disambiguation Using Neural Word Embeddings. In *BioNLP 2015*, pages 171–176, Beijing, China, 2015. doi: 10.18653/v1/W15-3822.

[42] Y. Wu, J. C. Denny, S. Trent Rosenbloom, R. A. Miller, D. A. Giuse, L. Wang, C. Blanquicett, E. Soysal, J. Xu, and H. Xu. A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD). *Journal of the American Medical Informatics Association*, 24(e1):e79–e86, 2017. ISSN 1527974X. doi: 10.1093/jamia/ocw109.

[43] H. Xu, P. D. Stetson, and C. Friedman. Methods for Building Sense Inventories of Abbreviations in Clinical Notes. *JAIMA*, 16(1):103–108, jan 2009. ISSN 10675027. doi: 10.1197/JAMIA.M2927.

[44] H. Xu, P. D. Stetson, and C. Friedman. Combining Corpus-derived Sense Profiles with Estimated Frequency Information to Disambiguate Clinical Abbreviations. In *AMIA Annu Symp Proc*, volume 2012, pages 1004–1013, 2012. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540457.

[45] H. Xu, Y. Wu, N. Elhadad, P. D. Stetson, and C. Friedman. A new clustering method for detecting rare senses of abbreviations in clinical notes. *Journal of Biomedical Informatics*, 45(6):1075–1083, dec 2012. ISSN 15320464. doi: 10.1016/J.JBI.2012.06.003.

[46] H. Yu, G. Hripcsak, and C. Friedman. Mapping Abbreviations to Full Forms in Biomedical Articles. *Journal of the American Medical Informatics Association*, 9(3):262–272, may 2002. ISSN 10675027. doi: 10.1197/JAMIA.M0913.

[47] W. Zhou, V. I. Torvik, and N. R. Smalheiser. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, 22(22):2813–2818, nov 2006. ISSN 13674803. doi: 10.1093/BIOINFORMATICS/BTL480.

Index of Abbreviations

Abbreviation	Definition	First appearance
Conv.	conventional features	p. 35
EM	expectation maximization	p. 12
MAX_SBE	maximum of each SBE dimension	p. 15
MRR	mean reciprocal rank	p. 33
NLP	natural language processing	p. 4
pmi	point-wise mutual information	p. 11
SBE	surrounding-based embedding	p. 14
Subst.	pmi-weighted substitution lemmas	p. 35
TCRS	Tight Clustering for Rare Senses	p. 12
WMD	word mover’s distance	p. 8
WSD	word sense disambiguation	p. 5
WSI	word-sense induction	p. 5

A Heuristic Abbreviation Identification

Disclaimer to not self-plagiarize: the heuristic was formulated during my research topics, so the text in this Appendix originates from my research topic report.

The heuristic aimed to capture abbreviations based on word formation that is unique to Dutch abbreviations. Unfortunately, writers of healthcare reports were not bound to grammar, and could use grammatically incorrect punctuation and capitalization. This meant that the heuristic produced false positives. These were filtered out by only considering abbreviations that occur frequently, as described in Section 4.

The heuristic recognized the following word formations of a token as unique to abbreviations:

- a word without vowels (e.g. *ghz*, *VVT*);
- a span of single letters separated by periods (e.g. *i.v.m.*);
- a word followed by a period that is not at the end of a sentence;
- a word that is fully capitalized.

These word formations were based on heuristics formulated for English abbreviations [23, 26], and characteristics of Dutch abbreviations stipulated by Taalunie[3], a renowned organization focussed on policy and development of the Dutch language.

[3] `https://woordenlijst.org/leidraad/17`

B Inner-Outer-Sense Retrieval Healthcare Abbreviations

It was argued that abbreviation sense retrieval via pattern-matching would not work for healthcare reports, contrary to biomedical papers. Since this statement was dependent on the corpus, this hypothesis was tested through a small empirical analysis. The analysis entailed pattern-matching for inner-outer pairs that contained abbreviations. The inner-outer pairs were pieces of text where parentheses were used. The inner was the text between parentheses, and the outer consisted of the 5 tokens before the parentheses, for example: outer-"*focus has been FF*", and inner-"*family first*". These pairs were then manually inspected to identify abbreviation-sense pairs, such as "*FF: family first*" in the example inner-outer pair.

The inner-outer-sense retrieval was conducted for 60 abbreviations of 2 or 3 letters per healthcare sector, totaling roughly 4,250 inner-outer pairs for analysis. For the elderly care sector, a sense was found for 7 abbreviations, of which 6 could be pattern-matched with the abbreviation. For the disabled care sector, a sense was found for 10 abbreviations, of which 9 could be pattern-matched. For the mental healthcare sector, a sense was found for 25 abbreviations, and multiple for 4 abbreviations, of which 29/32 could be pattern-matched.

As expected, many of the inner-outer pairs did not include senses, namely 98.6%. Mostly, the text between parentheses after an abbreviation was just a sidenote that had nothing to do with the sense of the abbreviation. To automate the extraction, the extracted pairs could be filtered by applying the validity checks presented by [3]. However, considering that few abbreviation senses were found using inner-outer extraction, it seemed best to discard this method in favor of word-sense induction. A big concern was that Dutch words contained far more compound words than English words, which would make pattern-matching more tricky. Inner-outer-sense retrieval would thus be very prone to false positives, aside from already being unable to retrieve senses for most abbreviations.

C Error Analysis Supplement

This error analysis supplement contains large tables that show the results at a low level. They show how well the best WSI method, *Subst. + TCRS fixed*, and candidate sense ranking worked for each abbreviation sense. In Section 6.2, the senses are categorized to show the reader which type of senses are most troublesome and what kind of errors they bring about to WSI and candidate sense ranking.

Table 21 and 21 show additional metrics for WSI per sense. Table 22 and 22 show the rank of each sense using the two semantic similarity measures for ranking and random ranking.

Abbr-eviation	Sense / Domain	#Anno-tations	Spread	#Noise	Precision	Recall	F1-score
	Elderly care						
hb	hemoglobine	52	4	11	1.000	0.692	0.818
	huisbezoek	4	1	2	0.500	0.500	0.500
ab	activiteitenbegeleider	2	2	0	1.000	0.500	0.667
	antibiotica	57	7	23	1.000	0.228	0.371
dd	dagdienst	10	2	0	1.000	0.900	0.947
	de dato	11	3	0	0.778	0.636	0.700
	de die	23	7	1	1.000	0.435	0.606
	dienstdoende	4	3	0	1.000	0.500	0.667
	differentiaaldiagnose	4	3	0	1.000	0.250	0.400
hh	herhalingen	17	4	4	1.000	0.529	0.692
	hoofd hals	1	0	1	0.000	0.000	0.000
	huishoudelijke hulp	37	9	22	0.833	0.135	0.233
wv	waarvoor	47	10	14	1.000	0.298	0.459
	wijkverpleegkundige	9	4	4	1.000	0.222	0.364
	Mental health care						
cm	casemanager	3	0	3	0.000	0.000	0.000
	centimeter	1	1	0	1.000	1.000	1.000
	contactmoment	49	12	25	1.000	0.082	0.151
gg	geen gehoor	5	1	1	0.235	0.800	0.364
	gegeven	1	0	1	0.000	0.000	0.000
	gegevens	2	0	2	0.000	0.000	0.000
	groepsgenoot	46	9	15	0.765	0.283	0.413
gz	gezins	1	0	1	0.000	0.000	0.000
	gezonde volwassene	5	2	3	1.000	0.200	0.333
	gezondheidszorg	33	6	8	0.917	0.333	0.489
bw	beschermd wonen	58	13	23	1.000	0.121	0.215
	bewindvoerder	1	1	0	0.333	1.000	0.500

TABLE 20: Low-level results for WSI using *Subst. + TCRS fixed* (part 1/2). The *#annotations* is the number of annotated occurrences of this sense in the ground truth. The *spread* is the number of clusters that had any occurrences of this sense. The *#noise* is the number of annotated occurrences that are considered noise by TCRS. The *precision*, *recall* and *F1-score* are denoted for the cluster with the highest F1-score for this sense.

Abbr- eviation	Sense / Domain	#Anno- tations	Spread	#Noise	Precision	Recall	F1-score
	Disability care						
hb	hemoglobine	3	1	2	1.000	0.333	0.500
	huisbezoek	38	9	12	1.000	0.184	0.311
gb	Gigabyte	3	0	3	0.000	0.000	0.000
	geen bijzonderheden	14	4	2	1.000	0.500	0.667
	gezinsbegeleider	7	2	1	1.000	0.571	0.727
ab	activiteitenbegeleiding	22	6	8	1.000	0.227	0.370
	ambulant begeleider	5	2	3	1.000	0.200	0.333
	antibiotica	4	3	1	1.000	0.250	0.400
dd	Donald Duck	2	0	2	0.000	0.000	0.000
	de dato	16	4	5	1.000	0.438	0.609
	de die	25	4	3	1.000	0.720	0.837
	dienstdoende	2	1	1	0.333	0.500	0.400
	differentiaaldiagnose	9	3	0	1.000	0.778	0.875
ib	individueel begeleider	51	8	13	1.000	0.392	0.563
	intern begeleider	2	1	1	0.167	0.500	0.250
hh	herhalingen	15	6	2	1.000	0.267	0.421
	huishoudelijke hulp	29	4	21	0.750	0.103	0.182

TABLE 21: Low-level results for WSI using *Subst. + TCRS fixed* (part 2/2).

Abbre- eviation	**Sense / Domain**	**Relaxed WMD rank**	**Sentence similarity rank**	**Baseline rank**
	Elderly care			
hb	hemoglobine	1646	2122	1168
	huisbezoek	2237	2001	1168
ab	antibiotica	45	61	558
	activiteitenbegeleider	-	-	-
dd	dienstdoende	14	39	2534
	de die	4099	3441	2534
	de dato	-	-	-
	differentiaaldiagnose	-	-	-
	dagdienst	7	9	2534
hh	huishoudelijke hulp	296	263	1184
	hoofd hals	-	-	-
	herhalingen	12	27	1184
wv	waarvoor	37	66	878
	wijkverpleegkundige	73	89	878
	Mental healthcare			
cm	contactmoment	845	797	1049
	centimeter	31	119	1049
	casemanager	1105	1065	1049
gg	groepsgenoot	440	195	4384
	geen gehoor	138	310	4384
	gegevens	2812	2363	4384
	gegeven	2178	3784	4384
gz	gezondheidszorg	1171	1414	1896
	gezonde volwassene	2318	2501	1896
	gezins	3414	3445	1896
bw	beschermd wonen	1909	1154	1997
	bewindvoerder	242	431	1997

TABLE 22: Low-level results for candidate sense ranking annotated occurrences (part 1/2). The senses without ranks are those that do not occur among candidate senses.

Abbre- eviation	Sense / Domain	Relaxed WMD rank	Sentence similarity rank	Baseline rank
	Disability care			
hb	hemoglobine	-	-	-
	huisbezoek	4494	4207	2408
gb	geen bijzonderheden	209	483	1578
	Gigabyte	-	-	-
	gezinsbegeleider	1862	1709	1578
ab	ambulant begeleider	1777	1243	1114
	antibiotica	362	493	1114
	activiteitenbegeleiding	560	960	1114
dd	de die	6829	6884	4654
	de dato	-	-	-
	Donald Duck	-	-	-
	differentiaaldiagnose	-	-	-
	dienstdoende	31	44	4654
ib	individueel begeleider	1044	1038	551
	intern begeleider	105	337	551
hh	huishoudelijke hulp	889	1764	2202
	herhalingen	3	11	2202

TABLE 23: Low-level results for candidate sense ranking annotated occurrences (part 2/2).