Word Sense Disambiguation of Clinical Abbreviations with Hyperdimensional Computing

Sungrim Moon, PhD[1], Bjoern-Toby Berster[2], MS,
Hua Xu, PhD[1], Trevor Cohen, MBChB, PhD[1, 3],
[1]The University of Texas School of Biomedical Informatics at Houston, Houston, TX;
[2]Drchrono, Mountain View, CA; [3]National Center for Cognitive Informatics and Decision Making in Healthcare, Houston, TX

Abstract

Automated Word Sense Disambiguation in clinical documents is a prerequisite to accurate extraction of medical information. Emerging methods utilizing hyperdimensional computing present new approaches to this problem. In this paper, we evaluate one such approach, the Binary Spatter Code Word Sense Disambiguation algorithm, on 50 ambiguous abbreviation sets derived from clinical notes. This algorithm uses reversible vector transformations to encode ambiguous terms and their context-specific senses into vectors representing surrounding terms. The sense for a new context is then inferred from vectors representing the terms it contains. One-to-one BSC-WSD achieves average accuracy of 94.55% when considering the orientation and distance of neighboring terms relative to the target abbreviation, outperforming Support Vector Machine and Naïve Bayes classifiers. Furthermore, it is practical to deal with all 50 abbreviations in an identical manner using a single one-to-many BSC-WSD model with average accuracy of 93.91%, which is not possible with common machine learning algorithms.

Introduction

Ambiguity (one word with multiple possible meanings) is very common in clinical text, especially for clinical abbreviations (including both acronyms and other abbreviated words) (1-2). Ambiguous words are often used to convey essential medical information (3), so correctly interpreting the meaning of an ambiguous term, referred to as Word Sense Disambiguation (WSD) (4), is important. Consequently, automated WSD is a critical cornerstone for the development of high quality medical Natural Language Processing (NLP) systems (5). However, automatic interpretation of the correct meaning of a given word within a sentence is non-trivial, and remains one of the major challenges in medical NLP research (6-7). Many WSD methods have been proposed, including Machine Learning (ML)-based, knowledge-based, and hybrid approaches. Among them, supervised ML algorithms have shown excellent performance in WSD tasks. However, individual classifiers need to be built for each ambiguous term, which makes ML-based methods less practical. As one of the approaches to solve automatic WSD, there is a recently emerged distributional model named the Binary Spatter Code Word Sense Disambiguation (BSC-WSD) algorithm, which is based on the hyperdimensional computing paradigm (8). This algorithm has shown performance comparable to established ML approaches on a disambiguation test set derived from the biomedical literature (9), but has yet to be adapted to, or evaluated in the context of the task of disambiguating clinical terms.

This paper discusses the evaluation and refinement of the BSC-WSD algorithm for clinical abbreviation disambiguation with two main contributions: 1) We demonstrate that the BSC-WSD algorithm can achieve a better performance than the Support Vector Machine (SVM) and the Naïve Bayes (NB) algorithms for disambiguation of clinical abbreviations; and 2) We modify the BSC-WSD algorithm to take into account the orientation and distance of other terms with respect to the ambiguous term (i.e., occurrence to the left or to the right of the ambiguous term), achieving a better performance. Unlike the other approaches evaluated, a single BSC-WSD model can disambiguate all abbreviations in the corpus without complex parameter optimizations, indicating a more practical solution for clinical WSD.

Background

Supervised ML algorithms have been utilized to address the WSD problem in the clinical domain (10-11). In general, supervised ML achieves high performance when trained on enough annotated samples with well-distributed senses (12). It is well known that obtaining sufficient numbers of annotated examples for supervised ML creates a bottleneck (13), as it requires the effort and time of medical experts. The SVM and NB algorithms are commonly used classifiers with stable performance on clinical WSD tasks (10). However, no clearly superior ML algorithms have been identified for the resolution of ambiguous abbreviations to date (11,14). Another limitation of ML-based WSD methods is that an individual classifier is required to resolve each ambiguous word.

As inputs for supervised ML algorithms, various types of features from overlapping multi-disciplinary fields have been considered. Generally, these are categorized into domain knowledge-based, linguistic, statistical, and general document-level features from the fields of biomedical NLP, computational linguistics, statistics, and medicine (15). Domain knowledge-based features often rely on medical vocabularies and ontologies such as the Unified Medical Language System (UMLS) (16) and customized medical dictionaries. Linguistic features are derived from the patterns of human natural languages, and statistical features use general distributional information such as co-occurrence statistics. Lastly, clinical features often include clinical contextual information such as medical specialties or section header titles in clinical notes. Among various feature types, the statistical Bag-of-Words (BoW) feature, which considers the words surrounding an ambiguous term, has been broadly used for clinical WSD tasks (10-11).

Emerging approaches to modeling semantics have used reversible vector transformations to encode additional information about the target term, such as the relative position of other terms (17-18) and syntactic dependencies (19). In recent work, we have used a reversible vector transformation to encode the *sense* of an ambiguous term into *semantic vector* representations of neighboring terms. A vector representation that combines elemental vector representations for an ambiguous word (term) and one of its senses (meanings) is generated. This combined "word-sense" vector is encoded into the semantic vector representations for terms surrounding the ambiguous word. When a new context with the ambiguous term is encountered, applying the inverse of this transformation to the semantic vectors for the terms in this context results in the recovery of the (sense) vector representation for the context-appropriate sense of the term concerned. This approach, named the Binary Spatter Code WSD algorithm (BSC-WSD) in accordance with the underlying representational approach (20), showed comparable performance to existing supervised ML methods on a test set derived from the biomedical literature (9). However, it has yet to be evaluated for abbreviation disambiguation, or on clinical texts.

The Binary Spatter Code (BSC) (21) is one of a family of representational approaches collectively known as Vector Symbolic Architectures (VSAs) (22). VSAs were developed in response to Fodor and Pylyshyn's widely publicized critique of the inability of connectionist models of cognition to encode nested compositional structure (23). As such, they descend from Smolensky's tensor product-based approach (24), but offer considerable advantages in scalability on account of their utilization of reversible vector products that preserve the dimensionality of their component vectors. For example, the BSC uses pairwise exclusive OR (XOR) to combine high-dimensional binary component vectors, while Plate's widely-used Holographic Reduced Representations use circular convolution of either real or complex component vectors (25). VSAs have been applied to model a range of cognitive phenomena (for example (17,26,27)), by encoding information using vectors of high dimensionality as a representational unit. On account of this representational choice, it has been argued that VSAs and their operators represent a new computational paradigm, termed hyperdimensional computing (8), which provides human-like characteristics such as robustness to noise, approximate matching and analogical reasoning that are absent in traditional computational architectures. While a detailed account of this argument is beyond the scope of this paper, we refer the interested reader to Pentti Kanerva's excellent introduction to the subject for further details (8).

Recently, VSAs have been used to encode additional information into distributional models of semantics (17–19). Of particular relevance to the current work, Sahlgren and his colleagues used an alternative approach in which reversible vector transformation is achieved by permuting vector elements (18), to encode the relative position of terms in a sliding window. In some of these experiments, permutation of elemental vectors representing terms was used to indicate their orientation to the left or to the right of a target term. This approach produced the best results in a synonym test evaluation when compared with other ways of encoding relative position, suggesting that encoding orientation in this manner may be beneficial for distributional models of term semantics. Motivated by this finding, as well as the precedent for encoding orientation provided by the seminal Hyperspace Analog to Language model (HAL) (28) and recent findings that terms to the left of an ambiguous term may be of greater value for classification

(15), we introduce in this paper a novel variant of the BSC-WSD algorithm. This model uses permutation of elemental vectors to indicate their orientation with respect to a target term, with the hypothesis that this additional information will lead to enhanced performance. In addition, we evaluate the effect of weighting the contribution of surrounding terms to a context vector such that their contribution is inversely proportional to their distance from the target term, an approach that also has precedent in the HAL model (28).

In the section that follows, we describe our experiments in which we evaluate this novel variant of BSC-WSD against our original BSC-WSD implementation, as well as two widely used supervised ML methods, using a set of human-annotated ambiguous clinical abbreviations.

Method

Data set

The Natural Language Processing/Information Extraction (NLP/IE) (University of Minnesota) group released a dataset consisting of the most frequent acronyms and abbreviations encountered in clinical text (29). Moon et al. have created a dataset containing 50 ambiguous abbreviations with annotated samples (15), which were used in this study. Each acronym and abbreviation set consists of 500 annotated samples. Individual samples typically include several sentences that surround the primary sentence that contains the target acronym or abbreviation. For our experiments, we extracted the 500 sentences containing the targeted acronyms and abbreviations. Therefore, a total of 25,000 sentences (50 acronyms/abbreviations * 500 sentences) were used in this study. These 50 acronyms and abbreviations have a total of 267 senses, with an average of approximately five senses per acronym or abbreviation. Therefore, this set is both larger in size and more complex with respect to the number of senses than the NLM-WSD data set used to evaluate the BSC-WSD algorithm previously (9).

Features

BoW features based on sentences were utilized, because a BoW feature has been an effective and simple statistical feature (10-11) for sense disambiguation tasks as compared with other types of features such as semantic type information or part-of-speech tags. As a basic BoW feature, all occurring unique terms were considered at the sentence level. Pre-processing of free text features was identical across experiments, and accomplished using the indexing component Lucene (30), an open source search engine used to derive statistical features from each sentence. This was regarded as one feature set. As an alternative, words in the left word window preceding the target abbreviation and words in the right word window after the target abbreviation were considered as different features. This was done because the left window was shown to offer more utility for classification as compared with the right window in previous research (15). For example, when "CVP" is the target ambiguous term, the first and second appearance of the term "line" are considered as different features, considering the orientation/direction in the sentence below (the *left window* is shown in italics, and the right window in **boldface**).

> "*CV line was placed and his initial* CVP **rate was in the range of 4, and internal line has also been placed by anesthesiology**."

We encoded this feature using a novel permutation-based variant of the BSC-WSD algorithm that we will describe subsequently. We also evaluated additional features such as the section header information (a total of 1,510 unique section headers titles. Examples include "Chief Complaint", "History Of The Present Illness", "Impression/Plan", and "Laboratory Results") for local contextual information within in the clinical documents. In addition, we extracted Unified Medical Language System (UMLS) (16) Concept Unique Identifiers (CUIs) and semantic type information for these sentences. CUIs and semantic types were extracted from the sentence components surrounding the acronym/abbreviation in question using MetaMap (31) (version 2011) including all concepts extracted with a high confidence score (A cut-off score of 900 or 1000).

Algorithm

For the baseline, three fully supervised classification algorithms were evaluated. The performance of Majority Sense (taking the most frequently occurring sense in the training component of each pre-defined 10-fold cross validation set), NB, and SVM algorithms were utilized as implemented in the ZeroR, NaiveBayes, and LibSVM classes respectively in the open source Weka software package (32). These were then compared with variants of BSC-WSD, including a novel adaptation to encode orientation. In the case of the SVM, we used the C-SVC type and linear kernels with optimized parameters (cost and epsilon).

The BSC-WSD algorithm was implemented based on methods and classes provided by the Semantic Vectors software package (33–36). The algorithm is based on the Binary Spatter Code (BSC) (21), developed by Pentti Kanerva, which uses hyperdimensional binary vectors (dimension >= 10,000 bits) to represent both terms (or concepts) and the nature of the relationships between them. The algorithm proceeds as follows.

First, *elemental vector* representations are created for each ambiguous term and unique sense. We will refer to these elemental vectors as *E*(term) and *E*(sense), respectively. For example, for the sentence above we would anticipate generating the elemental vectors *E*(CVP) (representing the ambiguous term) and *E*(CVP|Central Venous Pressure) (representing the context-specific sense). Elemental vectors in the BSC are hyperdimensional binary vectors, initialized at random with a 50% chance of either a zero or one occurring at each dimension. The dimensionality of the space makes it highly probable that any two randomly constructed vectors will be orthogonal, or close-to-orthogonal, to one another (with orthogonal defined as Hamming distance = $\frac{dimension}{2}$ - for a detailed statistical analysis see (8,37)). This property makes the model robust, as it is highly unlikely that any two elemental vectors will be confused with one another, despite their being distorted during the training process. In order to ensure consistency across experiments, we employed a deterministic variant of this approach in which the random number generator is seeded with a hash function derived from the term in question, which is described in more detail in (38).

Given a context for training, the elemental vector for the ambiguous term and sense in this context are combined with one another to generate a *bound product*. We will refer to this binding operation using the symbol $\otimes$. With the BSC, binding is accomplished using pairwise exclusive OR (XOR), which is its own inverse. However, to maintain consistency with other hyperdimensional computing approaches we will use the symbol $\oslash$ to refer to the *inverse* of binding. The bound product representing the ambiguous term and its specific sense is added to the *semantic vectors* of the other terms that co-occur with the ambiguous term in this context (for example, other terms in the same sentence). These are initially zero, but acquire information as training proceeds. With the BSC, addition of binary vectors is accomplished by counting the number of 1's and 0's added in each dimension, and assigning a 1 to the *superposed product*, if the number of 1's is greater. In the case that the number of 1's and 0's are tied, 1 or 0 is randomly assigned. When this has occurred for all contexts, ambiguous terms, and senses, training is complete.

When a new context is encountered, the semantic vectors for the terms in this context are added together, and the inverse of the binding product is applied to this cumulative context vector. As shown with the symbolic representation below, we would anticipate the result of this operation approximating the vector representing the context-appropriate sense of the term concerned. In symbols, the key operations of the BSC-WSD are as follows:

Training: for every other term in the training context

$$S(term) \mathrel{+}= E(sense) \otimes E(ambiguous\ term)$$

Testing: for a newly encountered context

$$S(context) = \sum_{k=1}^{n} S(term^k)$$

$$S(context) \oslash S(ambiguous\ term) \approx E(sense)$$

The vector product $S(\text{context}) \oslash S(\text{ambiguous term})$ is compared with the vectors representing each sense of the term concerned, and the sense with the most similar vector representation to this product is selected.

BSC-WSD with 32,768 dimensions was applied with three different settings (orientation, log weighting and distance weighting). In some experiments, we modified the basic BSC-WSD algorithm to encode the direction of a term relative to the target term using permutation of the bound product during training, and reversing this permutation during testing (Orientation-based BSC-WSD in Table 1). The permutations utilized involved shifting every block of

64 bits one position to the right (P^{+1}) or left (P^{-1}). For terms to the left of the target term during training, $S(\text{term}) += P^{-1}\{E(\text{sense}) \otimes E(\text{ambiguous term})\}$ and during testing $S(\text{context}) += P^{+1}\{S(\text{term})\}$. For simple discrimination from the left orientation, the orientation of terms on the right was encoded by exclusion and no permutation was required.

We tested two other settings of the BSC-WSD algorithm. One differentiation involved weighting words in accordance with their statistical distributions across the corpus and distance from the target word (Weighted and D-Weighted BSC-WSD in Table 1). The other involved changing the mapping method used in the testing phase (One-to-one or One-to-all mapping of BSC-WSD in Table 1). In the first case, words the semantic context vectors constructed during the test phase are weighted in accordance with their local and global frequencies using the log-entropy weighting metric (39), and additions to the semantic term vectors of surrounding terms are weighted using the local "log" component of this metric during the training phase. Entropy-based BSC-WSD offers more weight to words with high local frequencies (the local "log" component), and less to words with high global frequencies (the global "entropy" component). In addition, additions during both phases are weighted depending on the distance between the term concerned and the target word. In other words, the closest word to the target acronym/abbreviation has a highest weight within a single context.

The log-entropy weighting for individual term was calculated as follows:

$$Global\ weight\ (i) = 1 + \sum_j \frac{p_{ij} log_2(p_{ij})}{log_2 n}\ where,$$

$$p_{ij} = \frac{Frequency\ of\ term\ i\ in\ document\ j}{Global\ frequency\ of\ term\ i}$$

$$Local\ weight\ (i,j) = log\ (1 + Frequency\ of\ term\ i\ in\ document\ j)$$

$$Log\ entropy\ (i,j) = \ Global\ weight\ (i) \times Local\ weight\ (i,j)$$

The distance weighting was calculated as follows:

$$Distance\ Weighting = \ Log\ entropy\ (i,j) * \left(\frac{1}{distance\ from\ term\ i\ to\ target\ term\ in\ document\ j}\right)$$

Another setting involved the sense-mapping method. The BSC-WSD algorithm is restricted to finding a sense within senses of the particular abbreviation only (One-to-one mapping of BSC-WSD in Table 1) in the test phase. In other words, the BSC-WSD algorithm can limit the scope of senses for categorization to those that are relevant to the individual target term, or use a single set of vectors containing all of the senses of all of the ambiguous terms in the set for convenient search. In the latter case, randomly occurring overlap between elemental vectors results in a small probability that the vectors representing the senses of other target terms will be retrieved instead of the vector representing the relevant sense of the target term. We would anticipate this noise leading to a small drop in performance, in exchange for the convenience of maintaining a common search space that addresses all ambiguous terms. When BSC-WSD deals with the senses of all ambiguous terms simultaneously, it can be considered as a one-to-all mapping method. On the other hand, if BSC-WSD constrains its scope to the senses of one target abbreviation, the classification task corresponds to the approach taken with other commonly applied methods of supervised ML, as it functions as a one-to-one mapping method. In practice, with the BSC-WSD the difference between these approaches is minimal, as one-to-one mapping is accomplished by restricting the search space in the test phase to those senses related to the term in question by incorporating a single "if" statement (if the term is stored with the sense vector concerned) without changing the training process. However, the one-to-one approach provides the grounds for a fair comparison with our baseline methods, as only the senses of the term in question are considered as possibilities for classification.

For the system evaluation, accuracy is reported in 10-fold cross-validation settings. Accuracy can be calculated as:

$$Accuracy = \frac{correctly\ classified\ samples * 100}{total\ number\ of\ samples}$$

Table 1. Best performance of baseline algorithms in comparison with different BSC-WSD settings

Word	Baseline			BSC-WSD			
	Majority Sense	NB	SVM (cost=1 & epsilon=0.5)	Oriented & Weighted (one-to-one)	Oriented & D-Weighted (one-to-one)	Oriented & Weighted (one-to-all)	Oriented & D-Weighted (one-to-all)
AB	69.00	96.20	96.80	95.80	96.40	95.80	96.40
AC	17.80	90.40	92.00	94.40	94.60	94.40	94.20
ASA	80.80	94.40	97.40	96.20	95.40	95.20	93.80
AV	74.80	95.00	97.40	96.60	98.00	96.60	98.00
AVR	76.20	95.60	94.80	94.40	95.80	94.40	95.00
BAL	91.40	95.60	95.80	93.00	93.20	92.40	92.40
BK	68.60	98.20	97.00	98.40	99.00	98.40	99.00
C&S	86.80	98.00	99.20	98.80	99.00	98.00	98.60
C3	42.00	95.00	94.80	97.60	97.00	97.60	96.80
C4	52.20	94.80	93.80	96.60	95.60	96.00	95.60
CA	78.20	95.40	95.80	90.00	93.80	89.80	93.20
CDI	24.80	95.40	97.40	95.40	97.00	95.40	96.60
CEA	88.80	95.60	95.40	93.60	95.00	93.60	94.60
CTA	79.20	96.20	96.60	97.80	97.00	96.80	96.20
CVA	55.60	96.80	97.80	97.20	95.80	96.60	95.20
CVP	87.20	97.20	94.80	95.40	94.80	95.40	94.20
CVS	91.40	89.00	90.00	91.00	91.40	90.40	90.20
DC	56.40	88.20	92.20	89.40	91.60	89.00	90.60
DIP	92.40	97.20	98.00	96.80	97.40	96.60	97.20
DM	57.20	94.80	95.60	95.40	96.00	95.40	95.40
DT	67.20	90.60	93.20	93.80	93.60	93.00	93.00
ER	89.60	98.60	98.60	95.40	97.80	95.20	97.60
FISH	89.80	99.60	98.40	99.20	99.20	99.20	99.20
IM	92.20	97.80	98.40	99.20	99.20	99.20	99.00
IR	78.80	97.60	96.60	98.20	98.60	98.20	98.40
IT	45.00	88.00	87.80	88.20	91.80	87.80	91.20
IVF	61.60	95.00	91.80	94.60	95.00	94.40	94.20
LE	69.00	95.20	93.60	93.20	93.00	93.20	92.40
MP	35.80	68.20	66.40	65.80	68.80	65.40	68.60
MR	62.80	94.60	91.40	94.80	95.40	94.60	95.40
MSSA	83.60	95.40	94.80	94.80	94.80	94.80	94.80
NAD	75.40	94.40	93.20	93.20	95.00	90.40	90.60
OP	61.60	94.60	95.40	95.40	95.80	68.20	90.40
OTC	93.80	91.60	96.80	96.40	97.20	95.60	95.60
PA	42.40	93.00	93.40	89.00	94.40	89.00	94.20
PAC	55.00	92.60	93.00	93.60	95.40	93.60	95.20
PCP	58.80	89.80	89.40	86.40	91.20	86.40	91.00
PDA	72.20	91.60	88.40	90.00	92.20	90.00	91.60
PE	81.60	93.40	99.20	98.40	98.40	98.20	98.40
PR	50.40	96.40	96.40	95.00	96.00	94.80	95.80
RA	78.80	90.40	90.40	88.80	90.20	88.00	89.60
RT	67.20	91.60	93.80	92.40	93.20	92.20	92.20
SA	74.60	95.40	94.80	93.80	95.40	93.40	94.80
SBP	83.40	95.80	96.00	96.00	95.80	95.80	95.20
SMA	70.60	90.20	88.40	85.60	88.60	85.60	88.60
T1	35.00	91.40	90.00	93.00	94.40	93.00	94.40
T2	45.40	89.80	88.20	84.80	90.40	84.80	90.40
T3	53.60	93.40	91.40	92.40	91.40	92.00	91.00
T4	84.80	96.60	95.20	95.20	95.60	95.00	95.20
VBG	59.80	94.20	95.40	95.60	96.00	93.80	94.20
Average	67.81	93.72	93.85	93.52	94.55	92.65	93.91

* Oriented: Orientation-based BSC-WSD
* Weighted: Entropy-based BSC-WSD
* D-Weighted: Entropy-based BSC-WSD with distance weighting
* One-to-one: one-to-one mapping method of BSC-WSD
* One-to-all: one-to-all mapping method of BSC-WSD

Result

When considering BoW features only without orientation, SVM classifiers (with optimized parameter cost as 1 and epsilon as 0.5) offer the best performance with an average accuracy of 93.85%, as compared with NB classifiers' average accuracy of 93.72% across all 50 acronyms and abbreviations. NB classifiers show better sense disambiguation than Majority Sense classifiers, which have an average accuracy of 67.81%. The entropy-based BSC-WSD algorithm with distance weighing with sentence-level BoW features has an average accuracy of 92.56% (one-to-one mapping) and 91.61% (one-to-all mapping), although in 13 of 50 abbreviations (10 cases for one-to-one BSC-WSD and 3 cases for one-to-all BSC-WSD) it shows the best accuracy over all classifiers when considering BoW features exclusively.

With regard to orientation/direction of BoW, the average accuracy of SVM classifiers (with optimized parameter cost as 1 and epsilon as 0.3) and NB classifiers slightly decreased in performance to 93.54% and 92.79%, respectively. Table 1 represents the performance of the novel permutation-based variant of the BSC-WSD algorithm that takes into account the orientation/direction, and at times the distance of the BoW features relative to the target term. The best average accuracy is 94.55% using entropy-based distance weighting and one-to-one mapping with the BSC-WSD algorithm. In the case of entropy-based distance weighting and one-to-all mapping, the BSC-WSD algorithm has an accuracy of 93.91%. The BSC-WSD algorithm with one-to-one mapping shows 31 of the best individual performances among 50 sets over all classifiers. In case of the BSC-WSD algorithm with one-to-all mapping, it shows 9 of the best individual performances.

The results of the BSC-WSD algorithm (oriented, entropy-based distance weighting and one-to-one mapping) exceed the best performance of the SVM and NB algorithms. According to the Wilcoxon signed-rank test between the best results from SVM and NB algorithms (without orientation and with optimized parameter cost as 1 and epsilon as 0.5) and individual results from the oriented and weighted one-to-one mapping BSC-WSD algorithm, these improvements are statistically significant ($p = 0.008$ between BSC-WSD and SVM, $p = 0.003$ between BSC-WSD and NB). 23 of the 50 abbreviation sets show the best performance with this BSC-WSD algorithm over all classifiers. There is no statistically significant difference between the one-to-all BSC-WSD algorithm (oriented, entropy-based with distance weighting) and the best SVM algorithm (without orientation and with optimized parameter cost as 1 and epsilon as 0.5), though the performance of BSC-WSD is still superior with this mapping.

Different settings of the BSC-WSD algorithm present independent improvements of performance. The performance with orientation-based, log entropy-based, and log entropy-based with distance weighting BSC-WSD were improved by 1.06%, 1.60%, and 1.70% respectively as compared with the basic BSC-WSD algorithm (which had an average accuracy of 90.86%).

Discussion and limitation

This paper demonstrates the application of the supervised BSC-WSD algorithms to disambiguate 50 abbreviations in clinical texts with various settings. By taking into account the orientation/direction of BoW features relative to the target term, our novel permutation-based variant of the BSC-WSD algorithm shows significantly better performance than SVM and NB algorithms. Furthermore, the one-to-many mapping-based BSC-WSD algorithm can disambiguate 50 acronyms and abbreviations with reasonably high accuracy. This offers an advantage over common supervised ML algorithms, where an independent classifier must be trained for each term.

Several other features were applied in an attempt to improve performance. However, performance was not improved in the disambiguation task. Mapped CUIs information and semantic information utilizing MetaMap resulted in a slight deterioration of the overall performance of BoW features. These phenomena may be explained by the fact that these features are originally based on BoW features. Adding additional information did not appear to add valuable information. Another possible explanation is that limited clinical term coverage (2), inconsistences and ambiguous concepts within the UMLS (5,40) negatively affected the overall performance. Combining BoW features with section header information from clinical texts also did not improve the performance. Different expressions of word form or inconsistent levels of hierarchy for section header information may introduce additional noise into the disambiguation process. Therefore, we are investigating the use of controlled terminologies of section headers such as SecTag (41).

As we considered as a context the sentence containing the acronym concerned, the presence of clinical texts containing many fragmented sentences may be a potential limitation (42) in our study. Clinical "sentences" are often

ungrammatical and some of them lack periods. Moreover, formatting of note style may be different depending on the preference of each individual author. For example, one physician may prefer to enter a space between a section header and the remaining context, while another may tend to use tabulation to distinguish them. These incomplete sentence structures and different styles of writing cause difficulties for detection of consistent sentence boundaries, and these inconsistent sentence boundaries may damage the extraction of minimal contextual information for one sentence. However, this limitation would affect all algorithms tested.

Finally, there is a trade-off with respect to one parameter, the dimensionality used for the BSC-WSD algorithm. Performance remained strong at lower dimensionality also, with performance improvements over NB and SVM retained with vectors of 4,096 bits (the average accuracy of 94%). This reduction in dimensionality improves computational efficiency, as the average time for training scales at a rate that is linear to the dimensionality of the vectors (approximately 84 seconds per fold at 32K at 8 seconds per fold at 4K in our experiments).

Conclusion

This paper presents the application of the BSC-WSD algorithm to disambiguate acronyms and abbreviations in the clinical domain. The BSC-WSD algorithm without any parameter optimization shows competitive performance with common supervised ML algorithms. This is consistent with previous evaluations using a smaller test set derived from the biomedical literature. In addition, we developed a novel permutation-based variant of the algorithm that considers the orientation/direction and distance of BoW features with respect to the target term, further improving the performance of the algorithm. The best performance significantly outperforms the best performance obtained with the SVM and NB algorithms. Furthermore, the BSC-WSD algorithm creates one model for all 50 abbreviations to deal with sense disambiguation, presenting a convenient alternative to other supervised ML algorithms. While overall accuracy was similar across algorithms, we note that the performance across examples of the baseline algorithms was more strongly correlated with the performance of the majority sense approach (r=0.43 and 0.44 for NB and SVM respectively, as compared with r=0.33 (one-to-one mapping) and 0.31 (one-to-all mapping) for BSC-WSD). This suggests that the BSC-WSD approach may offer an advantage when the training examples for each sense are evenly distributed. The BSC-WSD algorithm is conceptually very different from the baseline approaches. Rather than defining a classifier for each ambiguous term, the information required to disambiguate all of the ambiguous terms encountered is dispersed across a set of semantic vectors representing other terms in the corpus. When a new context is encountered, a vector similar to the elemental vector used to encode the relevant sense is extracted from the superposition of the semantic vectors for the terms in this context. The fact that this approach appears to perform effectively in a manner that is complementary to established approaches suggests it represents a promising new avenue for WSD research.

Acknowledgements

The authors would like to thank The Natural Language Processing/Information Extraction (NLP/IE) (University of Minnesota) group for dataset. This work was supported by National Library of Medicine grant (NLM R01LM010681), National Cancer Institute grant (NCI R01CA141307), as well as US National Library of Medicine grant (R21LM010826-01), Encoding Semantic Knowledge in Vector Space for Biomedical Information Retrieval. The authors thank Buzhou Tang and Yonghui Wu for their insightful comments.

References

1. Stetson PD, Johnson SB, Scotch M, Hripcsak G. The sublanguage of cross-coverage. Proc AMIA Symp. 2002;742–6.
2. Xu H, Stetson PD, Friedman C. A study of abbreviations in clinical notes. AMIA Annu Symp Proc. 2007;821–5.
3. Walsh KE, Gurwitz JH. Medical abbreviations: writing little and communicating less. Arch. Dis. Child. 2008 Oct;93(10):816–7.
4. Pakhomov S. Semi-supervised Maximum Entropy based approach to acronym and abbreviation normalization in medical texts. Proceeding of the 40th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics; 2002. p. 160–7.

5. Friedman C, Liu H, Shagina L, Johnson S, Hripcsak G. Evaluating the UMLS as a source of lexical knowledge for medical language processing. Proc AMIA Symp. 2001;189–93.
6. Schuemie MJ, Kors JA, Mons B. Word sense disambiguation in the biomedical domain: an overview. J. Comput. Biol. 2005 Jun;12(5):554–65.
7. Kaplan A. An experimental study of ambiguity and context. Mechanical Translation. 1955;2:39–46.
8. Kanerva P. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. Cognitive Computation. 2009;1(2):139–59.
9. Berster B-T, Goodwin, Caleb, Cohen, Trevor. Hyperdimensional Computing Approach to Word Sense Disambiguation. Proc. AMIA Annu Symp. 2012.
10. Joshi M, Pakhomov S, Pedersen T, Chute CG. A Comparative Study of Supervised Learning as Applied to Acronym Expansion in Clinical Reports. AMIA Annu Symp Proc. 2006;2006:399–403.
11. Savova GK, Coden AR, Sominsky IL, Johnson R, Ogren PV, De Groen PC, et al. Word sense disambiguation across two domains: biomedical literature and clinical notes. J Biomed Inform. 2008 Dec;41(6):1088–100.
12. Xu H, Markatou M, Dimova R, Liu H, Friedman C. Machine learning and word sense disambiguation in the biomedical domain: design and evaluation issues. BMC Bioinformatics. 2006 Jul 5;7:334.
13. Xu H, Stetson PD, Friedman C. Methods for Building Sense Inventories of Abbreviations in Clinical Notes. J Am Med Inform Assoc. 2009;16(1):103–8.
14. Liu H, Teller V, Friedman C. A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation. J Am Med Inform Assoc. 2004;11(4):320–31.
15. Moon S, Pakhomov S, Melton GB. Automated Disambiguation of Acronyms and Abbreviations in Clinical Texts: Window and Training Size Considerations. AMIA Annu Symp Proc. 2012 Nov 3;2012:1310–9.
16. NIH. Unified Medical Language System [Internet]. Available from: http://www.nlm.nih.gov/research/umls/
17. Jones MN, Mewhort DJK. Representing word meaning and order information in a composite holographic lexicon. Psychological Review. 2007;114:1–37.
18. Sahlgren M, Holst A, Kanerva P. Permutations as a Means to Encode Order in Word Space. Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08), July 23-26, Washington D.C., USA. 2008;
19. Basile P, Caputo A, Semeraro G. Encoding syntactic dependencies by vector permutation. Proceedings of the EMNLP 2011 Workshop on GEometrical Models of Natural Language Semantics, GEMS. 2011 p. 43–51.
20. Kanerva P. The spatter code for encoding concepts at many levels. 1994.
21. Kanerva P. Binary spatter-coding of ordered K-tuples. Artificial Neural Networks—ICANN 96. 1996;869–73.
22. Gayler RW. Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience. In Peter Slezak (Ed.), ICCS/ASCS International Conference on Cognitive Science. Sydney, Australia. University of New South Wales.; 2004. p. 133–8.
23. Fodor JA, Pylyshyn ZW. Connectionism and cognitive architecture: A critical analysis. Cognition. 1988 Mar;28(1-2):3–71.
24. Smolensky P. Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial intelligence. 1990;46(1-2):159–216.
25. Plate TA. Holographic Reduced Representation: Distributed Representation for Cognitive Structures. Stanfpord, CA.: CSLI Publications; 2003.
26. Plate TA. Analogy retrieval and processing with distributed vector representations. Expert systems. 2000;17(1):29–40.
27. Eliasmith C, Thagard P. Integrating structure and meaning: A distributed model of analogical mapping. Cognitive Science. 2001;25(2):245–86.
28. Burgess C, Livesay K, Lund K. Explorations in Context Space: Words, Sentences, Discourse. Discourse Processes. 1998;
29. Natural Language Processing/Information Extraction (NLP/IE), University of Minnesota. Clinical Abbreviation Sense Inventory [Internet]. Available from: http://purl.umn.edu/137703
30. The Apache Software Foundation. Lucene Core [Internet]. http://lucene.apache.org/
31. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp. 2001;17–21.
32. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explor. Newsl. 2009 Nov;11(1):10–8.
33. Widdows, D, Cohen, T, DeVine, L. Real, Complex, and Binary Semantic Vectors. To appear in: QI'12. Proceedings of the 6th International Symposium on Quantum Interactions. Paris, France; 2012.
34. Widdows D, Cohen T. The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on. p. 9–15.

35. Widdows D, Ferraro K. Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. Sixth International Conference on Language Resources and Evaluation (LREC 2008). 2008.
36. Berster B-T. Binary Spatter Code - Word Sense Disambiguation [Internet]. Available from: https://github.com/tobyberster/BSP-WSD.
37. Kanerva P. Sparse distributed memory. Cambridge, Massachusetts: The MIT Press; 1988.
38. Wahle M, Widdows D, Herskovic JR, Bernstam EV, Cohen T. Deterministic Binary Vectors for Efficient Automated Indexing of MEDLINE/PubMed Abstracts. AMIA Annu Symp Proc 2012;2012:940–9.
39. Martin DI, Berry MW. Mathematical Foundations Behind Latent Semantic Analysis. Handbook of Latent Semantic Analysis. 2007.
40. Rindflesch TC, Aronson AR. Ambiguity resolution while mapping free text to the UMLS Metathesaurus. Proc Annu Symp Comput Appl Med Care. 1994;240–4.
41. Denny JC, Spickard A 3rd, Johnson KB, Peterson NB, Peterson JF, Miller RA. Evaluation of a method to identify and categorize section headers in clinical documents. J Am Med Inform Assoc. 2009 Dec;16(6):806806
42. Long WJ. Parsing Free Text Nursing Notes. AMIA Annu Symp Proc. 2003;2003:917