Learning multiple distributed prototypes of semantic categories for named entity recognition

Aron Henriksson

Department of Computer and Systems Sciences,
Stockholm University,
164 07 Kista, Sweden
Email: aronhen@dsv.su.se

Abstract: The scarcity of large labelled datasets comprising clinical text that can be exploited within the paradigm of supervised machine learning creates barriers for the secondary use of data from electronic health records. It is therefore important to develop capabilities to leverage the large amounts of unlabelled data that, indeed, tend to be readily available. One technique utilises distributional semantics to create word representations in a wholly unsupervised manner and uses existing training data to learn prototypical representations of predefined semantic categories. Features describing whether a given word belongs to a certain category are then provided to the learning algorithm. It has been shown that using multiple distributional semantic models, each employing a different word order strategy, can lead to enhanced predictive performance. Here, another hyperparameter is also varied – the size of the context window – and an experimental investigation shows that this leads to further performance gains.

Keywords: distributional semantics; semantic space ensembles; random indexing; named entity recognition; electronic health records; de-identification.

Reference to this paper should be made as follows: Henriksson, A. (2015) 'Learning multiple distributed prototypes of semantic categories for named entity recognition', *Int. J. Data Mining and Bioinformatics*, Vol. 13, No. 4, pp.395–411.

Biographical notes: Aron Henriksson received his BSc in Computer Science from Royal Melbourne Institute of Technology in 2008, his MSc in Information Systems from Royal Institute of Technology in 2010. He is currently a PhD candidate at Stockholm University. His research interests include natural language processing and machine learning.

This paper is a revised and expanded version of a paper entitled 'Generating features for named entity recognition by learning prototypes in semantic space: the case of de-identifying health records' presented at the 'IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2014)', 2–5 November 2014, Belfast, UK.

1 Introduction

Learning from high-dimensional and sparse data such as text is challenging and, in a supervised learning setting, requires substantial amounts of labelled data. Creating large amounts of labelled text data for every problem, domain and language is, however,

prohibitively expensive. It is therefore important to develop techniques that reduce the required amount of training data or, by the same token, enable learning of high-performing predictive models in situations where access to labelled data is limited. One possible remedy that has been well explored is to provide additional features to the learning algorithm by deriving distributed word representations from a large unlabelled corpus in a completely unsupervised manner. Such features help to reduce the sparsity in the labelled training data, which in turn can improve the generalisation accuracy of the learned predictive models. These types of approaches belong to a family of semi-supervised methods and have been shown to yield improved predictive performance on tasks including named entity recognition (Miller et al., 2004; Turian et al., 2010; Jonnalagadda et al., 2012).

Recently, the ideas underlying ensemble methods – i.e., combining multiple classifiers to make better predictions (Dietterich, 2004) – has been extended to models of distributional semantics, which capture lexical meaning on the basis of word co-occurrence information and is typically modelled in vector space (Turney and Pantel, 2010), allowing the semantic similarity of words to be quantified by calculating the distance between their vector representations. Several such semantic spaces, constructed over different types of corpora and with different hyperparameters, were shown to lead to improved predictive performance on a synonym extraction task (Henriksson et al., 2014b).

This notion has also been exploited in the context of named entity recognition in clinical text (Henriksson et al., 2014a). The method that was proposed uses a small amount of available instances for a given named entity class to learn a prototypical vector representation in semantic space, defined as the column-wise median of the instances' semantic vectors. Binary features that describe whether a given word belongs to that semantic category are then provided to the learning algorithm, where the feature values are determined by calculating the (cosine) distance and ascertaining whether they are below or above a given threshold, set to maximise F_1-score on the training set. In that study, combining multiple semantic spaces, each employing a different strategy for handling word order, led to improved predictive performance. Here, this notion is extended to create larger ensembles of semantic spaces, wherein another hyperparameter is also varied, namely the size of the context window in which co-occurrences are counted. It is shown that, by learning multiple distributed prototypes from a larger set of semantic spaces, further gains in predictive performance can be obtained, indicating that the learning algorithm is benefiting from the more holistic view of the data that this effectively provides. A number of follow-up analyses confirm that the semantic spaces are indeed providing diverse representations.

1.1 Named entity recognition in clinical text

Named entity recognition (NER) concerns the ability to recognise references to entities of certain predefined semantic categories in free-text. This ability is a key enabler of accurate information extraction, which has grown in importance with the inexorably growing amounts of digitised data. One application area for information extraction that is receiving considerable attention at the moment is healthcare, where great amounts of data

are now being stored as a result of the increasing adoption of electronic health records (EHRs). Since the majority of this data is in the form of text, information extraction and other natural language processing (NLP) methods need to be adapted to this particular domain. This is especially challenging due to the properties of clinical text: formal grammar is typically not complied with, while misspellings and non-standard shorthand abound (Allvin et al., 2011). Testament to the growing importance of domain-adapted NER systems are the many shared tasks and challenges that have been organised in recent years (Uzuner et al., 2010; Uzuner et al., 2011; Pradhan et al., 2014; Pradhan et al., 2015). However, most of the existing NER modules that are used in clinical NLP systems, such as MedLEE (Friedman, 1997), MetaMap (Aronson and Lang, 2010) and cTAKES (Savova et al., 2010), are rule-based – i.e., with hand-crafted rules – and thus rely heavily on comprehensive medical dictionaries. The trend is, however, increasingly moving in the direction of machine learning, with state-of-the-art clinical NER systems being primarily based on predictive models (De Bruijn et al., 2011; Tang et al., 2013; Zhang et al., 2014).

1.2 Distributional semantics

Word representations used in semi-supervised approaches to NER can be obtained with models of distributional semantics. Distributional semantics is a computational approach to modelling the meaning of natural language that is based on the observation – and captured in the distributional hypothesis (Harris, 1954) – that words with similar meanings tend to appear in similar contexts. Models of distributional semantics have primarily been used to create (semantic) vector representations of words, which have proved useful in a wide array of NLP tasks (Turney and Pantel, 2010). In recent years, distributional semantics has been leveraged also in the biomedical (Cohen and Widdows, 2009) and clinical (Henriksson, 2013) domains.

It has been shown that the predictive performance can be improved further by combining multiple semantic spaces, either by deriving the semantic vectors from different types of corpora or by changing the parameters of the models (Henriksson et al., 2014a; Henriksson et al., 2014b). Although different distributional semantic models have slightly different hyperparemeters, the definition of context is common to all and affects the properties of the semantic space (Sahlgren, 2006). An important distinction exists, for instance, between *syntagmatic* and *paradigmatic* relations, and which one is modelled depends on the context definition that is employed. The former holds between words that co-occur (e.g., {car, engine, road}) and is characterised by the size of the context region, while the latter holds between words that do not themselves co-occur but share neighbours (e.g., synonyms like {car, automobile}). Context is usually defined as a (sliding) window that is symmetric around the focus word. The size of the context window has also been shown to play an important role in contrasting different semantic relations (Lapesa et al., 2014), and the optimal window size tends to be task-dependent (Lapesa and Evert, 2014). For the task of extracting medical synonyms from large corpora, it has been shown that combining semantic spaces constructed with different hyperparameters, including window size, can lead to improved performance (Henriksson

et al., 2014b). For NER, using multiple semantic spaces constructed with different strategies for handling word order was shown to lead to improved performance compared to using only a single semantic space (Henriksson et al., 2014a).

2 Methods and materials

The initial version of the proposed semi-supervised approach (Henriksson et al., 2014a), as well as its extension, both presuppose the availability of two resources: (1) an annotated (named entity) corpus and (2) an unannotated corpus. While the annotated corpus may be relatively small, the unannotated corpus should preferably be much larger and in the same domain. The method essentially consists of the following steps:

1 Learning multiple distributed prototypes for each semantic category.

2 Generating features for the instances (words) based on their distance in semantic space to each of the prototype vectors.

3 Applying an appropriate learning algorithm to the annotated corpus with a feature set that includes the generated features.

The core of the method is in the first two steps, which concern the provision of semantic features to the learning algorithm with the use of distributed word representations. The focus of this study is, moreover, primarily on the first step, where a large set of distributed prototypes are learned for each semantic category, resulting, however, in a larger number of features in the second step, the use of which are evaluated in the third and final step. In addition, a number of follow-up analyses are conducted in order to gain further evidence of the benefit of semantic space ensembles, as well as insights into the effects of model hyperparameters on the resulting semantic spaces.

2.1 Learning multiple distributed prototypes

A distributed prototype vector is an abstract representation of a target (named entity) class. It is learned by exploiting the existing annotations to obtain their (distributed) representations in semantic space, which is constructed over a large, unannotated corpus. The prototype vector of a semantic category is then obtained by taking the centroid of the semantic vectors representing the category's annotated instances that occur above some threshold t in the unannotated corpus; here, t is set to a fairly large number: 100. Low-frequency terms are not included since the statistical foundation for their representation is weak, i.e., the observations of word usage are few. The centroid is defined as the median value of each dimension, as it was shown to lead to a better separation of classes compared to using the column-wise mean values (Henriksson et al., 2014a). This results in an abstract representation, i.e., one that does not correspond to an actual instance, which is otherwise often the case when calculating the centroid of a cluster. When employing a set of semantic spaces, a prototype vector is obtained for each semantic space and semantic category (Algorithm 1).

Here, a large set of semantic spaces are constructed over the unannotated corpus by varying two model hyperparameters: one concerns the strategy for handling word order in the context window, and the other is the size of the context window in which co-occurrences are counted. The semantic spaces are created with random indexing

(Kanerva et al., 2000), which is a scalable and computationally efficient model of distributional semantics. It creates a reduced-dimensional vector space in which the relative distances between vectors have been approximately preserved. In contrast to other dimensionality reduction techniques like singular value decomposition and the models of distributional semantics that depend on it, e.g., latent semantic analysis (Landauer and Dumais, 1997), it circumvents the need to construct an initial term-by-term matrix that is then reduced. Instead, pre-reduced vectors – in the sense that their dimensionality is much smaller than the size of the vocabulary – are incrementally populated with co-occurrence information.

Algorithm 1: Learning multiple prototype vectors for a semantic category

input: multiset W of seed words, set S of semantic spaces
output: set of n-dimensional distributed prototype vectors $P = \{\overrightarrow{p_1}, ..., \overrightarrow{p_{|S|}}\}$

```
for s ∈ S do
    for w ∈ W do
        v ← SemanticVector(w, s)
        /* append coordinate at position i in v to c_i                    */
        for i ← 1 to n do
            Append(v_i , c_i)
        end
    end
    /* get column-wise median values                                      */
    for i ← 1 to n do
        p_i ← Median(c_i)
    end
    /* append prototype vector from s to P                                */
    Append(p , P)
end
return P
```

In the construction of a semantic space with random indexing, there are two types of vectors: *index vectors*, which are used only in the construction phase, and *semantic vectors*, which represent the meaning of words and collectively make up the semantic space. Each unique word w_j in the vocabulary W is assigned an index vector $\vec{w}_j^i$ and a semantic vector $\vec{w}_j^s$ of dimensionality d, which is here set to 5000. The index vectors are static representations of the words that are approximately uncorrelated to each other. This is achieved by creating very sparse, ternary vectors that are randomly assigned a small number of non-zero elements (1s and –1s), in our case 50 (1%), with equally many 1s and –1s. A $\vec{w}_j^s$ – containing the distributional profile of the word w_j – is then the sum of all the index vectors of the words with which w_j co-occurs within a window of a certain size s. In contrast to the previous study (Henriksson et al., 2014a), where a single symmetric window size of 2 – two words to the left and right of the target word – was used, we here experiment with three different window sizes: 2^n, where $n = 1, 2, 3$ (2+2, 4+4, 8+8).

In this setting, word order within the context window is effectively ignored; however, it is also possible to encode word order information in the semantic vectors by permuting the elements of the index vectors on the fly before adding them to $\vec{w}_j^s$ (Sahlgren et al., 2008) – this has been shown to improve performance on various synonym extraction tasks (Sahlgren et al., 2008; Henriksson et al., 2013). The semantic vectors are referred to as **order vectors** when the elements in the index vectors are shifted according to their corresponding words' relative position to the target word: for a word that occurs two positions to the left of the target word, the elements of that word's index vector are shifted two positions to the left before adding the index vector to the semantic vector, and for a word that occurs one position to the right of the target word, the elements are shifted once to the right. The semantic vectors are referred to as **direction vectors** when the elements in the index vectors are shifted only one position depending on whether the corresponding word occurs to the left or the right of the target word. When the element vectors are not shifted at all, the semantic vectors are sometimes referred to simply as **context vectors**.

2.2 Generating distributional features

The prototype vectors are then used for generating features that describe the instances – which are here words or tokens – in the dataset. As in the previous study (Henriksson et al., 2014a), there is one binary feature per named entity class and semantic space, where the value is either True or False depending on whether the cosine similarity between the target word and the prototype vector is above a set threshold. The threshold is based on the pairwise distances between the annotated named entities of a certain semantic category and its corresponding prototype vector in a given semantic space. The threshold is set to maximise F_β-score on the training set, where the positive examples are the instances that belong to a certain semantic category and the negative examples are all other instances (equation 1).

$$arg\,max_{t \in \mathbb{V}} \left((1+\beta^2) \frac{P^{(t)} \cdot R^{(t)}}{(\beta^2 \cdot P^{(t)}) + R^{(t)}} \right), \tag{1}$$

where P is precision (true positives / true positives + false positives) and R is recall (true positives / true positives + false negatives); $\mathbb{V} = (0, 0.0001, 0.0002, ..., 1)$; β determines the weight that should be given to recall relative to precision. The lowest threshold is chosen that optimises the F_β-score. While the impact of using various β values has been studied previously (Henriksson et al., 2014a), a β value of 1 is used here, giving equal weight to precision and recall. In the same study, several different strategies for combining the features derived from multiple semantic spaces were compared; here, we employ the one that proved the most successful, in terms of yielding the highest F_1-score, namely retaining all the features that were generated by the multiple prototype vectors (Henriksson et al., 2014a).

2.3 Training named entity recognition model

In addition to the generated distributional semantic features, a set of orthographic and syntactic features are also generated. These features are commonly used for NER and are similar to the ones used in (Dalianis and Boström, 2012):

F_1: Is the token alphanumeric?

F_2: Is the token numeric?

F_3: Does the token have an initial capital letter?

F_4: What is the part-of-speech tag of the token?

F_5: What is the length of the token?

These features, in addition to the generated semantic features, are then provided to the learning algorithm together with the class labels. Following the standard approach to training a NER model, we cast the problem as a sequence labelling task, in which the goal is to find the best sequence of labels for a given input, i.e., the sequence of tokens in a sentence, which are described by various features. IOB-encoding of class labels is used, which indicates whether a token is at the beginning (B), inside (I) or outside (O) a given named entity mention. Here, the underlying learning algorithm is conditional random fields (CRF) (Lafferty et al., 2001), as implemented in CRF++ (Kudo, 2005), which is a popular choice for sequence labelling tasks. The power of CRF lies in its ability to model multiple variables that are dependent on each other – as they typically are in sequence labelling tasks – while exploiting large sets of input features. It achieves this by using an undirected probabilistic graphical model that, in contrast to, e.g., Hidden Markov Models (which is generative), is discriminative. Here, we use a linear-chain CRF that, in addition to being dependent on the input features, is also dependent on the previous and subsequent output variable. In the experiments described in this paper, the same hyperparameter settings as in previous studies involving the same dataset are used (Henriksson et al., 2014a; Dalianis and Boström, 2012): the L2-regularisation hyperparameter, which governs the balancing between underfitting and overfitting, is set to 5, and a symmetric window size of 2+2, which determines to what extent dependencies should be modelled between input features and output variables, is used.

2.4 Data source

The two corpora that are used in this study are subsets of the Stockholm EPR Corpus (Dalianis et al., 2009; Dalianis et al., 2012), which comprises health records from a wide range of healthcare units at Karolinska University Hospital in Stockholm, Sweden over a five-year period (2006–2010). This research has been approved by the Regional Ethical Review Board in Stockholm (permission number 2012/834-31/5). The two corpora are: (1) a small annotated PHI corpus and (2) a large unannotated corpus. The Stockholm EPR PHI Corpus (Dalianis and Velupillai, 2010) comprises 100 health records from five different clinics (Neurology, Orthopaedics, Infection, Dental Surgery, and Nutrition). This corpus originally contained 28 PHI classes that were annotated by three annotators;

see Velupillai et al. (2009) for a detailed description of the corpus creation process. A consensus-based gold standard was later derived from the original annotations after discussions between the annotators (Dalianis and Velupillai, 2010). This process included merging conceptually similar PHI classes, resulting in the following eight classes: *First Name*, *Last Name*, *Age*, *Health Care Unit*, *Location*, *Full Data*, *Date Part* and *Phone Number*. The version of the PHI corpus used in the following experiments contains a total of 198,821 tokens and 4321 annotated instances. The unannotated corpus, over which the semantic spaces are constructed, contains ten million clinical notes and approximately 169 million tokens (2.3 million types). In total, nine semantic spaces are constructed using the three strategies for handling word order – resulting in context vectors, direction vectors and order vectors – and three window sizes: 2+2, 4+4 and 8+8.

2.5 Experimental set-up

In the first and main experiment, two feature sets are provided to the learning algorithm and the predictive performance of the resulting NER models is compared: (1) using a combination of prototype vectors obtained with three different strategies for handling word order (context vectors, direction vectors and order vectors) and a single, 2+2, window size, and (2) using a combination of prototype vectors obtained with three different strategies for handling word order (context vectors, direction vectors and order vectors) and multiple window sizes (2+2, 4+4 and 8+8). We also compare the obtained results with two additional baselines: (1) using prototype vectors obtained with only a single semantic space, constructed using a 2+2 context window and direction vectors – these were previously shown to yield the best results (Henriksson et al., 2014a), and (2) using only the set of orthographic and syntactic features, without any semantic features. Finally, in order to ascertain that any performance gains obtained by the semantic space ensembles are not, in fact, the result of effectively giving more weight to the semantic features, we also evaluate a model that is given access to semantic features derived from a single semantic space (again, direction vectors with a 2+2 window) that are repeated as many times as there are semantic spaces in the largest ensemble.

A number of experiments and follow-up analyses are then conducted to investigate the contribution made by the semantic spaces from the perspective of word order strategy and window size, as well as to gather further evidence of the benefit of semantic space ensembles. We begin by first inspecting the threshold setting procedure in the respective semantic spaces to learn of any differences that may exist. We then systematically remove features from the large ensemble and study the impact this has on the predictive performance. Three semantic spaces – according to word order strategy or window size – are removed each time. Another form of analysis is to inspect the top-n nearest neighbours (NN) of the prototype vectors in semantic space, which is determined on the basis of their cosine similarity scores. We begin by qualitatively evaluating some of the prototype vectors by retrieving and inspecting their top-ten NN; to assess whether there are, in fact, differences between prototype vectors derived from different semantic spaces, we include the NN from two different semantic spaces for each prototype vector in the analysis. We then perform a quantitative analysis of the top-n NN of all prototype vectors, where n is set to 1000. There are several methods for comparing two ranked lists, which can be categorised into rank correlation methods and set-based methods

(Webber et al., 2010). Kendall's Tau (Kendall, 1938) belongs to the former and essentially measures the probability of two items being in the same order in the two ranked lists (equation 2):

$$\tau = \frac{C - D}{N}, \tag{2}$$

where C is the number of concordant pairs, i.e., the number of pairs for which the relative ordering is preserved in the two lists; D is the number of discordant pairs, i.e., the number of pairs for which the relative ordering is reversed; and N is the total number of pairs, $\frac{n(n-1)}{2}$, from a list with n items. The coefficient must be in the range [–1, 1], where a value of 1 indicates perfect agreement between the two rankings, a value of –1 indicates perfect disagreement between the two rankings, and a value of 0 indicates that the two rankings are independent. Here, we calculate the τ coefficient for all possible combination pairs of ranked lists within each semantic category, resulting in $\frac{9 \times 8}{2} \times 8 = 288$ comparisons. A problem with Kendall's Tau is, however, that it is unweighted, which means that the rank position of an item has no effect on the final similarity score. The property that is often desired is known as top-weightedness. Set-based metrics exist that satisfy the top-weightedness criterion. The basic idea is to calculate the fraction of content overlapping at different depths and then to return the average overlap. For two ranked sets, A and B, the average overlap score o between them can be defined as follows (equation 3).

$$o = \frac{\sum_{i=1}^{N} (|\{A_1, \ldots, A_i\} \cap \{B_1, \ldots, B_i\}| / i)}{N}, \tag{3}$$

where N is the length of both sets and the o coefficient must be in the range [0, 1]. This approach is naturally top-weighted, i.e., it gives more importance to items that are ranked highly, since observing a common item at a higher rank position contributes to all the lower-ranked intersections. Here, $N = 1000$. We calculate the average overlap scores for all possible combination pairs of ranked lists within each semantic category and average the scores across word order strategy and window size, respectively. This allows us to observe potential differences in the average overlap scores between different word order strategies, on the one hand, and between different window sizes, on the other.

In terms of evaluation, the considered performance metrics are precision, recall and F_1-score. Precision, which is also known as positive predictive value, is the fraction of predicted instances that are correctly labelled; recall, which is also known as sensitivity, is the fraction of positive instances that are predicted correctly; F_1-score is the harmonic mean between precision and recall (equation 4).

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

In all experiments, tenfold cross-validation is carried out. Performance scores are both micro- and macro-averaged: in the former, the metrics are calculated globally by counting the total numbers of true positives, false negatives and false positives, while, in the later, the metrics are calculated class-wise, after which their unweighted mean is taken, ignoring class imbalance.

3 Results

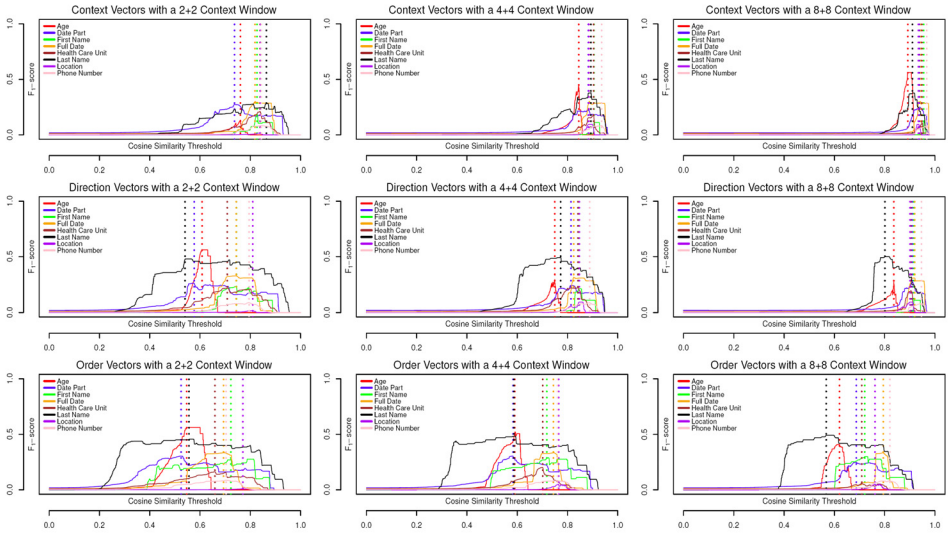
The first experiment showed that, in terms of macro-averaged precision, recall and F_1-score (Table 1), employing multiple window sizes (MWS) outperformed employing only a single window size (SWS). The same holds for the micro-averaged scores, save for precision, where that of SWS is slightly higher. The biggest improvement is observed for the *Location* class; for most other classes, the differences are generally rather small. Compared to using a single semantic space, however, the differences are somewhat bigger, with the micro-averaged F_1-score of MWS up 1.3 points. In comparison to not employing any semantic features, MWS obtains a macro-averaged F_1-score that is almost 3 points higher. In general, recall is benefiting to a greater extent than precision from the use of distributional semantic features. The results obtained when using repeated features from a single semantic space, which are lower than both ensembles (SWS and MWS), indicates that the improvement stems from combining multiple distinct semantic spaces and not simply from giving more weight to the semantic features. In fact, the results obtained with that model are even worse than when employing only a single semantic space (without repeated features).

Table 1 NER performance scores per PHI class for CRF models trained with orthographic and syntactic features, along with semantic features derived from three distributional semantic models that employ: a single window size (SWS) or multiple window sizes (MWS). The best results per PHI class and performance metric are made bold

Class	*Instances*	*Precision*		*Recall*		*F_1-score*	
		SWS	*MWS*	*SWS*	*MWS*	*SWS*	*MWS*
First Name	920	**95.322**	95.129	84.444	**84.903**	89.507	**89.677**
Last Name	927	95.004	**95.391**	90.358	**90.672**	92.554	**92.929**
Age	55	**74.333**	73.904	**65.762**	65.428	**69.218**	69.001
Health Care Unit	982	**81.769**	81.517	62.078	**62.882**	70.440	**70.926**
Location	147	86.150	**87.031**	50.322	**54.867**	62.493	**66.329**
Full Date	450	93.826	**94.150**	**95.204**	94.924	94.436	**94.464**
Date Part	706	**92.956**	92.370	**90.412**	89.938	**91.617**	91.093
Phone Number	134	94.512	**94.566**	**74.577**	72.171	**82.832**	81.342
Micro-average		91.612	91.509	81.063	81.370	85.996	86.123
Macro-average		89.234	89.257	76.645	76.973	81.637	82.568
Single semantic space (micro-average)		91.357		80.715		85.682	
Single semantic space (macro-average)		89.143		76.126		81.276	
Single semantic space, repeated (micro-average)		91.061		80.561		85.465	
Single semantic space, repeated (macro-average)		88.605		76.087		81.093	
Without semantic features (micro-average)		90.834		78.330		84.106	
Without semantic features (macro-average)		88.211		73.396		79.118	

Differences between the obtained prototype vectors can, in part, be observed from the threshold setting procedure, depicted in Figure 1. In general, the classes can be better separated with direction and order vectors, in comparison to context vectors, with higher F_1-scores obtained for most classes. It is also clear that the optimal thresholds are lower for direction and order vectors. There are, moreover, differences between the different window sizes: in general, the thresholds increase with larger window sizes, although it is not clear which window size results in a generally better separation of classes. For the *Age* class, for instance, a much higher F_1-score is obtained with a large window size (8+8) in the case of context vectors; however, the opposite is true in the case of direction vectors, where a small window size results in a higher F_1-score. It seems to be the case that using a larger window size is better for context vectors, while using a smaller window size is better for direction and order vectors.

Figure 1 An illustration of the setting of thresholds that maximise F_1-score for each PHI class and semantic space; thresholds are indicated by a dashed vertical line. N.B. The data is averaged over folds (see online version for colours)



The impact on predictive performance when removing a category of semantic spaces – either a strategy for handling word order or a window size – from the ensemble is shown in Table 2. The overall biggest drop in performance – for all considered metrics – is observed when removing the 2+2 window size, resulting in an almost two points lower macro-averaged F_1-score. Removing direction vectors has the biggest impact on performance when removing one of the different strategies for handling word order. The least impact on performance is observed when removing context vectors or the 4+4 window size. It should be noted, however, that the performance, in terms of F_1-score, is invariably reduced when removing any category of semantic spaces, demonstrating the advantage of semantic space ensembles.

Another means of investigating differences across various semantic spaces is to inspect the nearest neighbours (NN) of the prototype vectors that inhabit them. Examples of the top-ten NN for the prototype vectors of three semantic categories – *First Name*, *Location* and *Full Date* – are shown in Table 3. First of all, these examples serve to

demonstrate the feasibility of the approach: most, if not all, NN belong to the same semantic category as the corresponding prototype vector. It is also important to highlight that many NN appear high up in the lists without having been observed in the training data; these have been made bold. It is also clear that differences across semantic spaces do, indeed exist, which can be seen by merely inspecting the local neighbourhoods of the prototype vectors in semantic space. A clear difference can, for instance, be seen for the two prototype vectors for *Location*: in the semantic space with context vectors and a 8+8 window, the NN are all Swedish cities or suburbs, whereas in the semantic space with direction vectors and a 2+2 window, most of the NN are countries. Of the countries in this latter list, only two – *USA* and *Germany* – appeared in the training data; the rest were captured by sharing distributional properties with these countries and other locations. One last aspect that deserves attention is the many misspellings, e.g., the first name *Elisaeth* should be *Elisabeth* and the city *Örero* should be *Örebro*.

Table 2 Micro- and macro-averaged scores after removing different categories of semantic spaces from the multiple prototypes ensemble (MP), which is the MWS model from Table 1

		Precision		*Recall*		*F_1-score*	
		Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
Word Order	Multiple Prototypes (MP)	91.509	89.257	81.370	76.973	86.123	82.568
	MP – context vectors	91.491	89.744	81.128	77.102	85.983	82.221
	MP – direction vectors	91.389	88.799	80.922	76.793	85.823	81.640
Window Size	MP – order vectors	91.302	89.368	81.176	77.421	85.926	82.280
	MP – 2+2	91.095	88.123	80.186	75.877	85.277	80.875
	MP – 4+4	91.411	89.594	81.396	77.609	86.096	82.490
	MP – 8+8	91.331	89.077	80.955	76.853	85.814	81.774

Table 3 The top-ten nearest neighbours (NN) of prototype vectors inhabiting different semantic spaces. English translations are provided in square brackets when needed; words not available in the training set are made bold

NN Rank	$\overrightarrow{FirstName}$		$\overrightarrow{Location}$		$\overrightarrow{FullDate}$	
	Context, 2+2	*Order, 8+8*	*Context, 8+8*	*Direction, 2+2*	*Context, 4+4*	*Direction, 4+4*
1	maria	eva	skärholmen	hallen [*the hall*]	08	07
2	eva	maria	stockholm	**italien** [*italy*]	2008	08
3	anna	anna	**haninge**	**hudiksvall**	07	2008
4	annika	annika	**hudiksvall**	**portugal**	2007	2007
5	lena	karin	lund	**australien** [*australia*]	**nov08**	2006
6	**elisaeth**	**elisaeth**	gävle	usa	**jan09**	06
7	birgitta	kristina	**västerås**	tyskland [*germany*]	**dec08**	09
8	åsa	anders	**nynäshamn**	**frankrike** [*france*]	**okt08**	**nov08**
9	andreas	fredrik	örero	grekland [*greece*]	mars09	okt08
10	malin	lena	jakoserg	england	sept08	jan09

The differences between the NN of all prototypes for each respective semantic category were then evaluated in a more quantitative manner using the top-1000 NN. A comparison of all possible combination pairs of ranked lists resulted in τ coefficients ranging from –0.058 to 0.182, with an average τ coefficient of around 0.018. This indicates that the rankings are largely independent of each other. The average overlap scores across strategies for handling word order, on the one hand, and across window sizes, on the other, are shown in Table 4. The least amount of overlap is observed between context vectors and order vectors, followed by window sizes of 2+2 and 8+8. The highest degree of overlap is, on the other hand, observed between a window size of 4+4 and a window size of 8+8, followed by context vectors and direction vectors.

Table 4 Average overlap scores between different categories of semantic spaces

	Word Order				*Window Size*		
	Context	*Direction*	*Order*		*2+2*	*4+4*	*8+8*
Context	1	0.360	0.236	*2+2*	1	0.347	0.253
Direction	0.360	1	0.296	*4+4*	0.347	1	0.372
Order	0.236	0.296	1	*8+8*	0.253	0.372	1

4 Discussion

It was here shown that further improvements could be obtained for a semi-supervised approach to named entity recognition by employing a larger set of (nine) semantic spaces compared to a smaller set of (three) semantic spaces. In a previous study (Henriksson et al., 2014a), the smaller ensemble was created by exploiting three different strategies for handling word order in a distributional semantic framework; here, the larger ensemble was obtained by also utilising multiple window sizes. It has thus been shown that leveraging multiple semantic spaces constructed with different strategies for handling word order can outperform the use of only a single semantic space and a single word order strategy, and also that leveraging multiple semantic spaces constructed with different strategies for handling word order and different window sizes can outperform the use of multiple semantic spaces built with only a single window size. It remains to be seen, however, whether leveraging multiple semantic spaces constructed with different window sizes and a single word order strategy can outperform the use of multiple semantic spaces with a single window size and a single word order strategy. In this study, further evidence was gathered that indicates that the performance gains do, indeed, stem from the combination of distinct semantic spaces and not from, for instance, merely giving more weight to semantic features. This further strengthens the case for the potential of ensemble methods to be applied in the context of distributional semantics, which has shown promise in other tasks (Henriksson et al., 2014b). That said, the biggest difference in performance is observed with and without semantic features, although further improvements naturally become increasingly more difficult to obtain as performance increases.

The ensemble consists of multiple semantic spaces, each comprising a distinct set of prototype vectors for each semantic category. That the semantic spaces are different is key for the ensemble to be successful, as diversity is a key component of any ensemble

method (Dietterich, 2000). The diversity among the semantic spaces was here illustrated from several perspectives. The threshold setting procedure, illustrated in Figure 1, can, on the one hand, provide some indication as to which combination of hyperparameters is best able to separate the named entity classes; however, it is not entirely consistent in the sense that one configuration yields the highest F_1-scores for all classes, indicating that it may be better instead to combine the various prototype vectors. From a distributional semantic point-of-view, it is also interesting to note the impact of the combinations: clearly, a large window size yields better results in combination with context vectors, while a smaller window size yields better results in combination with direction and order vectors. A possible explanation for this is that the shifting of index vectors when using direction or order vectors requires a higher dimensionality when employing a larger window size, as the probability of index vectors sharing coordinates increases with these strategies for handling word order. On the other hand, it is clear that taking into account word order in some manner yields a better separation of classes, which confirms previous findings (Henriksson et al., 2014b). The outcome of the experiment wherein different categories of semantic spaces were removed also showed that the direction vectors made the biggest contribution to the performance of the ensemble; this was also shown to be the best single semantic space in the previous study (Henriksson et al., 2014b). Removing the 2+2 window size – the one employed in the previous study – caused a similar drop in performance, most probably as a result of removing the semantic spaces with direction and order vectors, as they seemed to have benefited most from employing a smaller window size. It is also interesting to note that employing a 2+2 window size yields good results for this particular task, as it has previously been shown to capture both synonymy (Henriksson et al., 2014b) and wider semantic categories (Skeppstedt et al., 2013) well.

When inspecting the nearest neighbours of the prototype vectors, it was also shown that notable differences do exist across semantic spaces. That the biggest differences were observed between context and order vectors is understandable, given the strict handling of word order in the latter: a co-occurrence event is then defined according to its exact position in relation to the target word, whereas, with context vectors, any co-occurrence of two words is counted as the same event. By the same token, it is not surprising that a larger difference was observed between a window size of 2+2 and 8+8 than between either 2+2 and 4+4 or between 4+4 and 8+8. The inspection of the top-ten nearest neighbours provided insights of a different kind, in addition to the simple fact that differences did indeed exist. While the capturing of instances that did not appear in the training data is essential for the method to be successful, it was interesting to see the number of misspellings that were successfully captured. These would not have been readily captured with dictionary-based approaches and doing so is essential when performing NER on clinical text, which is known to be noisy and replete with misspellings and ad-hoc abbreviations and acronyms (Allvin et al., 2011).

Although this approach is promising – as is the general notion of semantic space ensembles – it should in the future be evaluated on a number of datasets, preferably in different domains, as the approach is dependent neither on domain nor language. It should moreover be investigated if there are better ways of generating features with the use of prototype vectors – perhaps by circumventing the need for setting thresholds and not generating binary features. Finally, it would be possible to create even larger ensembles of semantic spaces by, for instance, using multiple corpora (as in Henriksson

et al., 2014b), or by employing additional models of distributional semantics, such context-predicting models, which have been shown to outperform context-counting models on a range of NLP tasks (Baroni et al., 2014).

5 Conclusions

We have extended a method for generating semantic features that may be exploited by the learning algorithm when training a named entity recognition model. A key feature of the method is that it leverages large amounts of unlabelled text data to supplement small amounts of training data by learning prototypical representations of named entity classes in (distributional) semantic space. The notion of semantic space ensembles is here extended to incorporate models built with different window sizes in addition to employing different strategies for handling word order, which is shown to yield further improvements in terms of predictive performance; the observed performance gains can to a large extent be attributed to diversity among the constituent semantic spaces. Methods that leverage large amounts of unlabelled data may reduce the amount of training data needed to obtain a certain level of performance within the paradigm of (semi-)supervised learning. This is of particular importance in specialised domains, such as healthcare, where annotated resources are typically scarce and often prohibitively expensive to create in large quantities.

Acknowledgements

This work was supported by the project High-Performance Data Mining for Drug Effect Detection at Stockholm University, funded by the Swedish Foundation for Strategic Research under grant IIS11-0053.

References

Allvin, H., Carlsson, E., Dalianis, H., Danielsson-Ojala, R., Daudaravicius, V., Hassel, M. et al. (2011) 'Characteristics of Finnish and Swedish intensive care nursing narratives: a comparative analysis to support the development of clinical language technologies', *Journal of Biomedical Semantics*, Vol. 2(S-3):S1.

Aronson, A.R. and Lang, F-M. (2010) 'An overview of metamap: historical perspective and recent advances', *Journal of the American Medical Informatics Association*, Vol. 17, No. 3, pp.229–236.

Baroni, M., Dinu, G. and Kruszewski, G. (2014) 'Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors', *Association for Computational Linguistics*, Vol. 1, pp.238–247.

Cohen, T. and Widdows, D. (2009) 'Empirical distributional semantics: methods and ical applications', *Journal of Biomedical Informatics*, Vol. 42, No. 2, pp.390–405.

Dalianis, H. and Boström, H. (2012) 'Releasing a Swedish clinical corpus after removing all words – de-identification experiments with conditional random fields and random forests', *Proceedings of BioTxtM 2012*, pp.45–48.

Dalianis, H. and Velupillai, S. (2010) 'De-identifying Swedish clinical text refinement of a gold standard and experiments with conditional random fields', *Journal of Biomedical Semantics*, Vol. 1, No. 6.

Dalianis, H., Hassel, M. and Velupillai, S. (2009) 'The Stockholm EPR corpus – characteristics and some initial findings', *Proceedings of the Symposium on Health Information Management Research*.

Dalianis, H., Hassel, M., Henriksson, A. and Skeppstedt, M. (2012) 'Stockholm EPR corpus: a clinical database used to improve health care', *Swedish Language Technology Conference*.

De Bruijn, B., Cherry, C., Kiritchenko, S., Martin, J. and Zhu, X. (2011) 'Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010', *Journal of the American Medical Informatics Association*, Vol. 18, No. 5, pp.557–562.

Dietterich, T.G. (2000) 'Ensemble methods in machine learning', *Multiple Classifier Systems*, Springer, pp.1–15.

Friedman, C. (1997) 'Towards a comprehensive medical language processing system: methods and issues', *Proceedings of the AMIA Annual Fall Symposium*, American Medical Informatics Association, p.595.

Harris, Z.S. (1954) 'Distributional structure', *Word*.

Henriksson, A. (2013) *Semantic spaces of clinical text: leveraging distributional semantics for natural language processing of electronic health records*, Licentiate Thesis, Stockholm University.

Henriksson, A., Conway, M., Duneld, M. and Chapman, W. (2013) 'Identifying synonymy between SNOMED clinical terms of varying length using distributional analysis of electronic health records', *AMIA Annual Symposium Proceedings*, pp.600–609.

Henriksson, A., Dalianis, H. and Kowalski, S. (2014a) 'Generating features for named entity recognition by learning prototypes in semantic space: the case of de-identifying health records', *IEEE International Conference on Bioinformatics and Biomedicine*, 2–5 November, Belfast, UK, pp.450–457.

Henriksson, A., Moen, H., Skeppstedt, M., Daudaravicius, V. and Duneld, M. (2014b) 'Synonym extraction and abbreviation expansion with ensembles of semantic spaces', *Journal of Biomedical Semantics*, Vol. 5, No. 6.

Jonnalagadda, S., Cohen, T., Wu, S. and Gonzalez, G. (2012) 'Enhancing clinical concept extraction with distributional semantics', *Journal of Biomedical Informatics*, Vol. 45, No. 1, pp.129–140.

Kanerva, P., Kristofersson, J. and Holst, A. (2000) 'Random indexing of text samples for latent semantic analysis', *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Vol. 1036.

Kendall, M.G. (1938) 'A new measure of rank correlation', *Biometrika*, pp.81–93.

Kudo, T. (2005) *Crf++: Yet another crf toolkit*. Software available online at: http://crfpp.sourceforge.net.

Lafferty, J., McCallum, A. and Pereira, F.C.N. (2001) 'Conditional random fields: probabilistic models for segmenting and labeling sequence data', *Proceedings of the 18th International Conference on Machine Learning*, pp.282–289.

Landauer, T.K. and Dumais, S.T. (1997) 'A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge', *Psychological Review*, Vol. 104, No. 2, p.211.

Lapesa, G. and Evert, S. (2014) 'A large scale evaluation of distributional semantic models: parameters, interactions and model selection', *Transactions of the Association for Computational Linguistics*, Vol. 2, pp.531–545.

Lapesa, G., Evert, S. and Schulte, S. (2014) 'imWalde. Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models', *Proceedings of the Third Joint Conference on Lexical and Computational Semantics* (** SEM 2014*), pp.160–170.

Miller, S., Guinness, J. and Zamanian, A. (2004) 'Name tagging with word clusters and discriminative training', *HLT-NAACL*, Vol. 4, pp.337–342.

Pradhan, S., Elhadad, N., Chapman, W., Manandhar, S. and Savova, G. (2014) 'Semeval-2014 task 7: analysis of clinical text', *SemEval 2014*, Vol. 199, No. 99, p.54.

Pradhan, S., Elhadad, N., South, B.R., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W. and Savova, G. (2015) 'Evaluating the state of the art in disorder recognition and normalization of the clinical narrative', *Journal of the American Medical Informatics Association*, Vol. 22, No. 1, pp.143–154.

Sahlgren, M. (2006) *The word-space model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*, PhD Thesis, Stockholm University.

Sahlgren, M., Holst, A. and Kanerva, P. (2008) 'Permutations as a means to encode order inword space', *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, pp.1300–1305.

Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C. and Chute. C.G. (2010) 'Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications', *Journal of the American Medical Informatics Association*, Vol. 17, No. 5, pp.507–513.

Skeppstedt, M., Ahltorp, M. and Henriksson, A. (2013) 'Vocabulary expansion by semantic extraction of medical terms', *Proceedings of the Symposium on Languages in Biology and Medicine* (*LBM*).

Tang, B., Cao, H., Wu, Y., Jiang, M. and Xu, H. (2013) 'Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features', *BMC Medical Informatics and Decision Making*, Vol. 13(Suppl 1):S1.

Turian, J., Ratinov, L. and Bengio, Y. (2010) 'Word representations: a simple and general method for semi-supervised learning', *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp.384–394.

Turney, P.D. and Pantel, P. (2010) 'From frequency to meaning: vector space models of semantics', *Journal of Artificial Intelligence Research*, Vol. 37, No. 1, pp.141–188.

Uzuner, Ö., Solti, I. and Cadag, E. (2010) 'Extracting medication information from clinical text', *Journal of the American Medical Informatics Association*, Vol. 17, No. 5, pp.514–518.

Uzuner, Ö., South, B.R., Shen, S. and DuVall, S.L. (2011) '2010 i2b2/va challenge on concepts, assertions, and relations in clinical text', *Journal of the American Medical Informatics Association*, Vol. 18, No. 5, pp.552–556.

Velupillai, S., Dalianis, H., Hassel, M. and Nilsson, G.H. (2009) 'Developing a standard for de-identifying electronic patient records written in Swedish: precision, recall and F-measure in a manual and computerized annotation trial', *International Journal of Medical Informatics*, Vol. 78, No. 12, pp.e19–e26.

Webber, W., Moffat, A. and Zobel, J. (2010) 'A similarity measure for indefinite rankings', *ACM Transactions on Information Systems*, Vol. 28, No. 4, p.20.

Zhang, Y., Wang, J., Tang, B., Wu, Y., Jiang, M., Chen, Y. and Xu, H. (2014) 'Uth_ccb: a report for semeval 2014–task 7 analysis of clinical text', *SemEval 2014*, pp.802–806.