

Unsupervised concept extraction from clinical text through semantic composition

Stéphan Tulkens*, Simon Šuster, Walter Daelemans

Computational Linguistics and Psycholinguistics (CLIPS) Research Center, University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium



ARTICLE INFO

Keywords:
Concepts
Clinical
UMLS
Unsupervised

ABSTRACT

Concept extraction is an important step in clinical natural language processing. Once extracted, the use of concepts can improve the accuracy and generalization of downstream systems. We present a new unsupervised system for the extraction of concepts from clinical text. The system creates representations of concepts from the Unified Medical Language System (UMLS®) by combining natural language descriptions of concepts with word representations, and composing these into higher-order concept vectors. These concept vectors are then used to assign labels to candidate phrases which are extracted using a syntactic chunker. Our approach scores an exact F-score of .32 and an inexact F-score of .45 on the well-known I2b2-2010 challenge corpus, outperforming the only other unsupervised concept extraction method. As our approach relies only on word representations and a chunker, it is completely unsupervised. As such, it can be applied to languages and corpora for which we do not have prior annotations. All our code is open-source and can be found at www.github.com/clips/conch.

1. Introduction

Concept extraction, also referred to as Named Entity Recognition (NER), is a task in which concepts or named entities are extracted from free text. It is widely used in general Natural Language Processing (NLP), and also an important task in the clinical domain, where it can help with the classification of diseases or recommendation of treatments by automatically detecting which concepts are present in, for example, an Electronic Health Record (EHR). The extraction of concepts can then improve the performance of downstream systems.

For the clinical domain specifically, however, several obstacles remain. One of these is access to clinical data: sharing EHRs, which contain sensitive patient information, presents hospitals with important practical and ethical issues regarding the research carried out on those records [1]. As a result, there are few annotated sources for concept extraction in the clinical domain.

Furthermore, because of the large variability in language use within and across clinical notes, the high volume of abbreviations and acronyms, specialized language use, and spelling mistakes [2], systems which are exposed to a single corpus of notes might not transfer well to other corpora of clinical notes [3]. For the task at hand, this problem is compounded by the fact that most publicly available corpora of patient records only contain records from a limited number of specialties, which constrains the number of concepts one encounters during

training, and hence the number of concepts one can predict with a supervised system. The i2b2-2010 corpus [4], for example, contains notes from three separate ICU wards from hospitals in the United States of America. Hence, any system which is trained on gold standard data from this corpus might not be as performant on data from other regions.

These factors, i.e. the relative shortage of data and the large variability within the available data, complicate the creation of accurate systems for the extraction of clinical named entities. Another confounding factor is that general, i.e. non-clinical, NER systems tend to focus on extracting proper names, such as organizations and specific persons [5,6]. Put in another way: clinical NER systems tend to focus on the extraction of information which is more related to facts than to the current status of the world than the information extracted by general NER systems. Therefore, techniques from general NLP might not transfer well to the clinical setting [7].

The goal of this paper is twofold: First, we make a case for using unsupervised techniques for concept extraction in the clinical domain, for the reasons stated above. Second, we provide a framework for the unsupervised extraction of clinical concepts through semantic composition, which we compare to CubNER [8], another method for unsupervised extraction of clinical concepts, and the current state of the art in unsupervised clinical concept extraction.

The method we present does not depend on gold standard labeled training corpora, and hence sidesteps the issue of the availability of

* Corresponding author.

E-mail addresses: stephan.tulkens@uantwerpen.be (S. Tulkens), simon.suster@uantwerpen.be (S. Šuster), walter.daelemans@uantwerpen.be (W. Daelemans).

labeled data, while not being fitted to any corpus in particular. Our approach starts from the idea that good semantic representations of words and phrases will provide us with enough leverage to accurately extract clinical terminology in the form of concepts. These concepts can then be used in downstream applications, as input to other classifiers, or as building blocks for other representations.

In previous work [9], we applied simple composition functions, such as averaging and addition, to distributed representations in order to solve the task of Word Sense Disambiguation (WSD) in a biomedical context. In WSD, the goal is to find the right sense of an ambiguous surface form, for example, to distinguish the two senses of the term “strep”, one of which denotes a streptococcus infection, and another which denotes the streptococcus bacteria. WSD is comparable to concept extraction in the sense that both tasks require the identification of unknown words. The main difference between the tasks is that in WSD the boundaries and identities of the ambiguous words are usually known, while concept extraction requires extraction of entities from running text. As NER is, in general, a more difficult task than WSD, the current system can be seen as an extension of the previous system.

The previous system worked as follows: for each of the ambiguous terms in the task, we created a representation of the context of this term by concatenating the right and left context words within a pre-specified window. We then replaced each word in this context by its distributed representation, and summed over this set of representations, creating a new distributed representation we called a context vector.

Concurrently we created a set of concept vectors from the Unified Medical Language System (UMLS®) Metathesaurus® ontology [10]. For each of the target concepts in the dataset, we took all descriptions, and performed composition by replacing each word by its representation and then summing. This led to what we called concept vectors.

We then calculated the cosine similarity between these concept vectors and the context vectors of ambiguous forms to retrieve the concept with the highest similarity between the ambiguous form and concept vector. The idea behind this process is that the local context in which a word occurs provides us with enough information to determine the particular sense of a word. In the current work, we use the same principle, but extend it to also include the phrase or word itself; in WSD the word or phrase itself precisely is the thing which needs to be disambiguated, and as such it can't be used as a reliable cue for disambiguation. In concept extraction, exactly the opposite is the case, and the phrase itself does provide a cue to its meaning.

In short, our approach uses unsupervised word representations, which represent the meaning of words in a single dense vector, and uses simple composition operators, e.g. addition or averaging, to create similar vector representations of multi-word units, such as phrases. In parallel, we use the same set of word representations and operators to compose descriptions, taken from the UMLS Metathesaurus, into higher-order concept representations. As in Tulkens et al. [9], because both the concept and phrase representations are created using the same technique, they are comparable, and can therefore be linked to one another on the basis of their distance in vector space.

2. Background

Our system is based on the idea that, when appropriately constructed, representations of phrases can be directly compared to representations of concepts. If we can find a way to construct concept representations that are situated in the same vector space as phrase representations, concept extraction reduces to matching each phrase representation with a concept representation. In this section, we will first review the literature on distributed representations of words and phrases, paying specific attention to the notion of compositionality, second, we will review how concept representations have been constructed. Finally, we give a short overview of other unsupervised techniques for NER.

2.1. Distributed representations

Recent work in the field of NLP has focused on creating vector representations of words which accurately represent the semantic properties of a word. This is usually achieved by exploiting what is known as the distributional hypothesis, which states that words which are similar in meaning are often used in similar contexts [11]. Hence, by representing words with vectors based on their co-occurrence patterns, words with similar meaning get similar D -dimensional vector representations, where D is usually a relatively small value, i.e., $50 < D < 500$. This approach has been particularly successful when implemented in a neural network which, instead of counting co-occurrences, learns word representations by predicting words from their context [12,13]. In any case, these word representations, having a fixed size, allow systems that use them to deal with very large vocabularies.

Given that we understand how to learn vectors for words, a simple way of learning phrase representations is to concatenate co-occurring words based on frequency [14]. For example, based on co-occurrence statistics, we might learn that “new” and “york” co-occur frequently, and we might then decide to concatenate them, and then learn a vector for “new_york” as a separate token. This compounding approach works quite well in the clinical setting, as shown by Henry et al. [15], but suffers from the downside that these tokens need to be detected and learned during training. Therefore, this method does not generalize to unseen compounds or phrases.

Moreover, while the problem of assigning semantic vectors to words and phrases on the basis of co-occurrence is well understood, the problem of constructing similar vectors for sentences or other higher-order structures is still largely unsolved [16]. This issue is intimately related to the compositionality of natural language; unlike words, whose semantics do not rely on the composition of individual letters, sentences and phrases get their meaning from the composition of the individual words making up that sentence [17]. An exception to this are compound idioms, such as “red tape” and “iron lung”, which do not neatly decompose, and are more than the sum of their parts [18].

As such, learning phrases as if they were regular words can only get us as far as our training data, as each phrase will need to have occurred in the training data in order to have a representation. Clearly, methods for modeling the semantics of higher-order units need to be able to mimic the compositionality of natural language to some degree.

Models that use parse trees to guide composition have been utilized to create accurate representations of the sentiment of a sentence [19–21] and paraphrases [22]. Note that applying a compositional function does not necessarily imply that the compositional function is required to respect syntactic principles, or include a model of syntax [23].

In this work, we take the latter route, and employ composition over words without taking into account the syntactic dependencies between words in that sentence. We refrained from using these syntactic dependencies because, in general, syntactic parses are expensive to construct, and might result in more noise when run on clinical text. Additionally, involving a syntactic parser makes the resulting system more language-specific, as parsers are not available for every language, especially in the clinical setting.

2.2. Distributed representations of concepts

Given the framework of distributed representations and phrases, we move on to describe work on creating vector representations of concepts. In order to function as representations in the context of concept extraction, these concept representations should be recoverable from the texts that describe them.

Hill et al. [24] create a model to map from descriptions of a concept to a word which denotes that concept, a task which is also referred to as reverse dictionary look-up. An example of reverse dictionary look-up is a mapping from “An animal of the African savannah with long legs and

highly elongated neck” to the word “giraffe”. To learn the mapping from descriptions to words they use a Long Short Term Memory (LSTM) [25], which is a gated variant of the Recurrent Neural Network (RNN) [26,27]. They compare ranking loss and cosine loss, among others, to predict the word representation of a word given its description, which is fed into the LSTM sequentially. To solve this task, the LSTM needs to learn to combine semantic representations in meaningful ways, and can thus be seen as a task in which an LSTM learns a closed composition function.

In the clinical domain, De Vine et al. [28] used a skipgram-like model to learn concept representations by first using Metamap[®], a supervised method, and then learning the distribution of these concepts from their co-occurrences.

Choi et al. [29] create distributed representations of UMLS Concept Unique Identifiers (CUIs), through the distribution of concepts within binned data gathered from an extremely large corpus of patient notes [30].

Similarly, Beam et al. [31] combine the two above works, and expand it to learn representations of about 108,000 concepts from three diverse corpora.

We argue that all these approaches are unsuitable for the current framework. The approach of Hill et al. [24] is unsuitable for our purposes, as one could argue that mapping from descriptions to words is not the same as mapping from phrases to concepts, as words may still be ambiguous in terms of the concept they represent. As such, learning to map from descriptions to concepts requires us to first solve the problem of getting appropriate vector representations for ambiguous concepts, which is exactly the task we are trying to solve in the current work. As such, we consider our work to be complementary to that of Hill et al. [24]. Similarly, the techniques for learning concept representations are not directly applicable to the current work, as they focus more on learning representations which might serve as input to other classifiers. The concept representations we use in the current work are instead focused on being extractable from text.

2.3. Clinical concept extraction

The work most related to ours is CubNER [8], a system for unsupervised concept extraction. CubNER uses several (pre-specified) semantic types from the UMLS semantic network to generate seed terms, which are then used to generate signature vectors for noun phrases extracted from the corpus. First, noun phrases are extracted by a chunker or parser. Following this, each of these noun phrases can be linked to one of the classes based on its similarity with each of the class representations. In contrast to our work, however, CubNER does not create task-agnostic concept representations, but instead tunes representations which are specific to a corpus, in this case the i2b2-2010 corpus [4]. As the methodology and corpora are close, we directly compare our system to CubNER in Experiment 3.

Also close to our approach is the work of Abacha and Zweigenbaum [32], in which an enhanced version of Metamap[®], a UMLS-based model, is used to extract concepts from the i2b2-2010 database. Specifically, they employ a tagger to extract syntactic features which are then utilized to limit the number of concepts extracted by Metamap[®].

In contrast to unsupervised extraction of concepts, of which the papers above were the only examples we could find in the clinical domain, there has been a lot of interest in the *supervised* extraction of concepts, most employing some sort of structured learning approach using Conditional Random Fields [4]. Recently, state-of-the-art NER system based on stacked bidirectional Recurrent Neural Networks with combined word and character-level features have proved to be performant in the supervised extraction of concepts from clinical and non-clinical text [33–36].

There are few resources for concept extraction in the clinical domain. Notable is the BluLab corpus, which was partially annotated by Kim et al. using the i2b2-2010 guidelines, but which was later

withdrawn from general use [3]. Other datasets are the SemEval 2014 [37] and 2015 [38] shared tasks, which included a NER component, in addition to other tasks. Both of these corpora consist of notes from the MIMIC corpus [39], annotated with disorder CUIs.

In general NLP, several systems for unsupervised NER have been proposed, including systems based on Adaptor Grammars [40], using the web as a corpus [41], syntactic rules [42], and parallel text [43].

3. Materials

3.1. Dataset

We test our approach on the i2b2-2010 challenge corpus [4], which is a dataset consisting of deidentified patient notes from three different institutions, namely Partners Healthcare, the Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center. While the original dataset contained 394 training reports and 477 test reports, the currently available dataset contains 170 training reports (73 from the Beth Israel Deaconess Medical Center and 97 from Partners Healthcare), and 256 test reports, the source distribution of which is not known. Each of these reports is annotated on the token span level using one of three labels: Test, Treatment and Problem. As such, the i2b2-2010 challenge corpus is a Named Entity Recognition (NER) sequence tagging problem.

As our method is unsupervised, we use the training documents from the Beth Israel Deaconess medical center as a development set. Because the text in the dataset was already tokenized, there was no need for any additional processing. We use the IOB tagging scheme [44] to represent which tokens belong to chunks. The corpus statistics are shown in Table 1.

3.2. Corpus

We used the MIMIC-III critical care database [39] to train our word representations. The MIMIC-III database consists of 53,423 Intensive Care Unit admissions of 46,467 distinct patients, and is the largest known publicly available corpus of patient notes to date. We pre-processed all free text patient records in the database by first lower-casing them and then using a regex-based tokenizer which also removed all of the de-identified markers, such as dates and time stamps. Our tokenized version of the MIMIC-III notes contains approximately 580 million words.

We then trained word representations on the resulting corpus of notes using FastText [45]. FastText is an extension of the well-known skipgram model [12], which projects words into a low-dimensional space by predicting the context of a word given the identity of the word. During the training, this causes words with similar contexts to be put closer together. The main innovation of FastText is that the model also estimates separate vectors for subword character *n*grams, which causes the model to be able to generalize beyond co-occurrence context. An example of this is the observation that words ending in “sarcoma” can be semantically grouped together regardless of their context.

We used the following hyperparameters: a window size of five, character *n*gram range of three to six, negative sampling with five negative samples, and a vector dimensionality of 320. This resulted in a set of 612,200 word representations. Note that we only use the word representations created by FastText, and do not include the OOV

Table 1

Corpus statistics for the train and test sets.

	# Words	Problem	Treatment	Test
Beth Israel	88,722	4187	3073	3036
Partners	60,819	2885	1768	1570
Test	267,249	12,592	9344	9225

estimation part of a trained FastText model in our experiments. We also used similar Skipgram representations, created with word2vec [12], but using the FastText representations improved performance.

3.3. UMLS Metathesaurus

Throughout this work, we used the 2015AB version of the UMLS Metathesaurus. We implement our own software for extracting concepts and their associated descriptions from the UMLS.¹

4. Overview of the method

Supervised sequence tagging usually entails simultaneous extraction and labeling of phrases in text. That is, most sequence taggers do not differentiate between extracting a phrase and assigning a label to this phrase.

In contrast, using an unsupervised method for a sequence tagging task naturally decomposes into a two-step process. In a first step, a set of phrasal units, commonly called the candidate set, is extracted from a set of documents. In a second step the candidate set is pruned and the remaining candidates are assigned labels, which can then be compared to the gold standard chunks. As these two steps are completely separate, we will describe them separately in the following sections. Fig. 1 shows a high-level overview of our system.

4.1. Step 1: phrase extraction

Like Zhang and Elhadad [8], we use a chunker to extract noun phrases from the texts of the i2b2 corpus, which we treat as candidate phrases for labeling. We used cTakes [46] to parse all documents, and extracted all noun phrases from these documents as candidates. Any phrases whose boundaries did not coincide with a word boundary were not extracted, because this would cause problems with the gold standard IOB alignment, which was on the token level. Because the i2b2 corpus was already tokenized, this only occurred in the case of time and dosage-related words such as “s/p” (status post), none of which were assigned a label. Because of this reason, we chose not to extract these chunks as candidates. Note that our use of phrases as candidates entails that we assume that concepts largely coincide with noun phrase boundaries. See Section 6.1 for an analysis of how warranted this assumption is.

This assumption was already investigated by Zhang and Elhadad [8], who concluded that, depending on the sub-corpus, between 3% and 5% of gold standard chunks did not have any overlap with any noun phrase, while around 50% of chunks completely overlapped with noun phrases, and around 45% of chunks either had partial overlap with a noun phrase, or were completely overlapped by a larger noun phrase. Because of this, they concluded that it is possible to use noun phrases extracted by a chunker to extract candidates.

However, because they used the Apache OpenNLP Chunker² we can not directly adopt their analysis; although cTakes uses the OpenNLP Chunker, it was retrained on a large set of clinical documents. Therefore, we perform an analysis on the output of our chunker in Section 6.1.

4.2. Step 2: candidate labeling

4.2.1. Semantic composition of phrases

After extracting the candidate set using cTakes, as described above, we compose the candidates into vectors using arithmetic composition over the vector representations of the words in each phrase. All extracted noun phrases were treated as candidates in the second step of

our analysis; we did not perform any additional pruning of the candidate set.

For each candidate, we extract all words within the phrase, and a window of N context words to each side of the phrase, obtaining three separate sequences of words. For each of these sequences, we replace all words in this sequence by their word representations, and use a composition function to obtain a D -dimensional vector, where D is the size of the original word representations.

More formally, for each sequence, we take the elementwise mean of all word representations in that sequence:

$$\text{mean}(W) = \frac{1}{|W|} \sum_{x \in W} x \quad (1)$$

where W is the set of representations in the sequence, x denotes an individual vector representation, and $| \cdot |$ denotes the cardinality operator

We apply Eq. (1) to the left context, the phrase and the right context to obtain three vectors. These three vectors are then composed again using an elementwise mean, obtaining a single vector:

$$c(W_l, W_p, W_r) = \frac{1}{3}(\text{mean}(W_l) + \text{mean}(W_p) + \text{mean}(W_r)) \quad (2)$$

That is, the components of candidate vector c are the mean of the components of the words in the left context W_l , the words in the phrase W_p , and the words in the right context W_r . As before, the resulting composition also has the same dimensionality as the word vectors.

Note that both of these formulas are actually just an application of an element-wise mean over the zeroth axis of a matrix of word or phrase representations. The composition function can therefore also be written as:

$$c(W_l, W_p, W_r) = \text{mean}(\text{mean}(W_l), \text{mean}(W_p), \text{mean}(W_r)) \quad (3)$$

More generally, our composition function can be rewritten as:

$$c(W_l, W_p, W_r, f_1, f_2) = f_2(f_1(W_l), f_1(W_p), f_1(W_r)) \quad (4)$$

where f_1 and f_2 are elementwise functions, and W_l , W_p and W_r are the word representations in the left context, phrase, and right context, respectively. Eq. (3) can be recovered from (4) by using mean (Eq. (1)) as both f_1 and f_2 .

As such, we experimented with a variety of different functions, and noted only a very small difference between using the elementwise averaging and addition functions. As the equations above show, we therefore use averaging as a function in all our experiments. Henry et al. [15] carried out a comprehensive evaluation of various elementwise composition functions and found that there was very little difference between using the element-wise mean and summation functions in composition of concepts [15].

Unlike Henry et al. [15], we also investigated elementwise multiplication as a composition function, as this has been shown to be successful in the context of count-based models [47,48]. This turned out not to work at all; any model which used multiplication obtained low scores on our development set in experiment 1. This is most likely due to the fact that the models produced by FastText represent their latent dimensions as fractions centered around the origin (i.e. the zero vector). As multiplications of fractions trend towards zero, multiplying several fractions almost always results in zero, or near-zero vectors, and therefore makes vectors indistinguishable from each other.

As an additional refinement, we weigh the words in both context windows by the reciprocal of their distance to the focus word, as follows:

$$\text{rec}_R(W) = \sum_i \frac{1}{i} W_i; \text{rec}_L(W) = \sum_i \frac{1}{N-i+1} W_i \quad (5)$$

where W again is a set of word representations and N is the number of representations in W . That is, given a context window, the first word

¹ The software is open-source, and available for use at www.github.com/clip/humumls.

² <https://opennlp.apache.org/>.

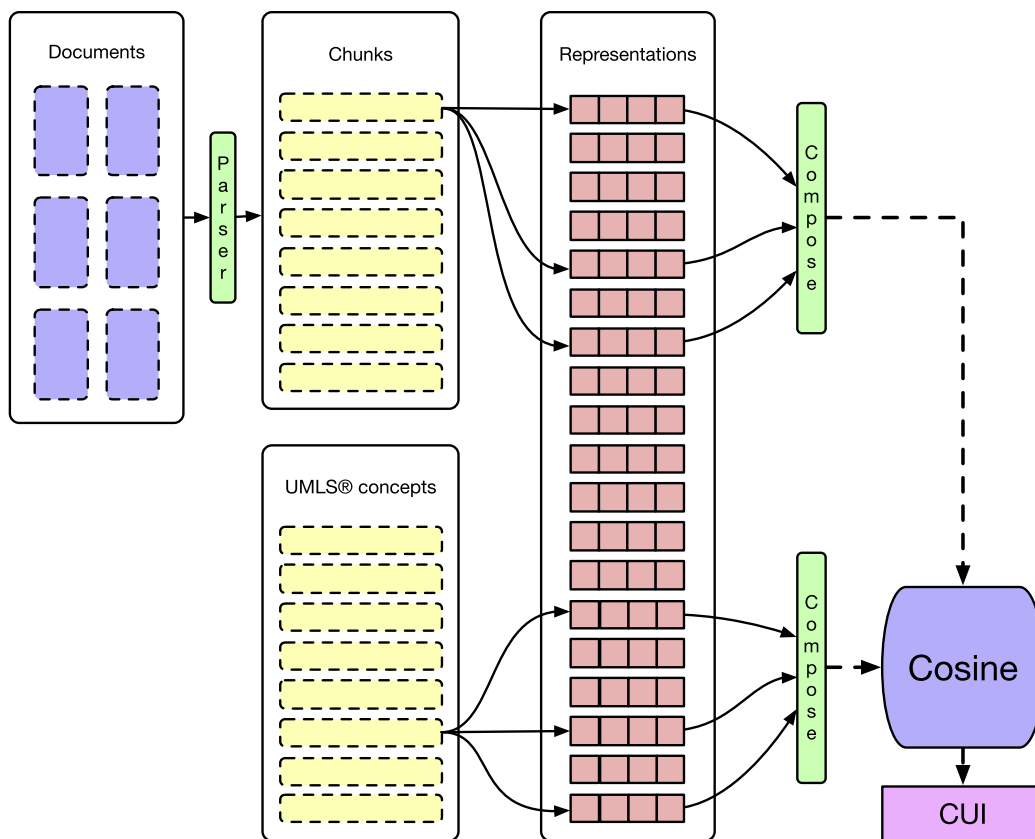


Fig. 1. An overview of the system. First, a chunker extracts phrases from a corpus, which are then composed into context vectors. In parallel, UMLS concepts are composed into concepts vectors. These are then compared, which allows the system to assign CUIs to individual chunks from the corpus.

will get a weight of $\frac{1}{1} = 1.0$, the second word will be weighted by $\frac{1}{2} = .5$, and so on.

Our final equation then becomes:

$$c(W_i, W_p, W_r) = \text{mean}(\text{mean}(\text{rec}_L(W_i)), \text{mean}(W_p), \text{mean}(\text{rec}_R(W_r))) \tag{6}$$

This improved performance dramatically, and also removes the need to manually specify window sizes; in practice, any word which is more than 10 words away will have such a low weight that its inclusion has a negligible impact on the overall semantic representation. The entire composition model is shown in Fig. 2.

Additionally, we also attempted to weigh the composition function using Inverse Document Frequency (IDF), as this has been shown to be effective in (biomedical) Information Retrieval-based settings [49]. Weighting word vectors by their IDF scores resulted in a slightly negative effect on performance on the development set in Experiment 1, and as such, we did not experiment with it further.

4.2.2. Semantic composition of concepts

As explained above, our method relies on a notion of similarity between candidate phrase vectors and concept vectors in assigning CUIs to phrases. As such, we again rely on composition functions to create

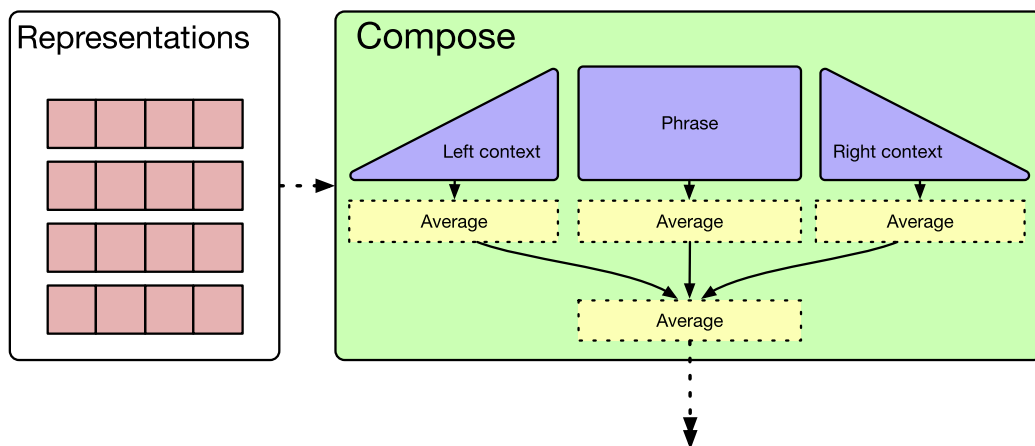


Fig. 2. A schematic overview of the composition step. The triangle shapes denote the reciprocal distance weighting function. Note that the representations refer to the representations of the words in the phrase.

vectors of concepts.

We extract all concepts from the UMLS Metathesaurus which have a definition in English, leading to a set of 178,343 concepts. To expand the coverage of our data, we also added the surface forms of the preferred terms to these concepts. We chose to restrict ourselves to concepts with definitions because of efficiency reasons; including all concepts with at least a single associated surface string drastically raised the number of concepts, from 178,343 to 3,221,699. In what follows, we refer to both definitions and surface strings as descriptions of a concept.

For each of the extracted concepts, we apply the functions detailed in equations f_1 and f_2 , as defined in the section above.

$$\text{concept} = \text{mean}(\text{mean}(W_0), \text{mean}(W_1), \dots, \text{mean}(W_n)) \quad (7)$$

where W_n denotes the set of word representations present in the n th description of the concept. Note that we do not apply any reciprocal weighting in the concept phrases, as there is no left or right context to speak of.

As we only use textual descriptions in our work, i.e. no relation information, the UMLS can be replaced by other dictionaries, e.g. language-specific terminology resources or other dictionaries.

4.2.3. Labeling concepts

Following Zhang and Elhadad [8], we assign each of the 178,343 extracted concepts one of the labels from the i2b2-2010 label set by exploiting the semantic group and semantic types in UMLS. Table 2 lists the semantic types or groups and their respective labels. Each of the concepts which has a link to one of these semantic types or groups gets assigned the respective label, while other concepts get assigned the label “np”. The distribution of labels is shown in Table 3. This labeling clearly shows a significant imbalance; most concepts are assigned the label “np”. Even if we only consider concepts assigned one of the three i2b2-2010 labels, we see that the number concepts assigned Treatment and Test is vastly lower than the number of concepts assigned Problem.

At the current juncture, it is important to note an important difference between the way our system uses these labels and the way they are used in CubNER. In CubNER, the semantic types and groups are used to create a single vector for each of the three labels in the i2b2 label set. This means that CubNER is inherently limited to assigning concepts with exactly these labels. For different corpora, another labeling strategy, based on domain knowledge, needs to be devised.

Our approach, in contrast, is more flexible and first assigns unique CUIs to each phrase, which are then converted to each of the labels in the i2b2-2010 dataset. This means that our system can assign any concept a CUI, which the user can then choose to convert to an appropriate label, depending on the context and user needs.

Furthermore, CubNER relies on TF-IDF filtering to remove concepts without clinical significance, while in our system we simply rely on the fact that concepts which are not relevant will get assigned “np” as a label. Note that this also removes the TF-IDF filtering threshold, an important free parameter in CubNER, from the model.

4.2.4. Assigning labels to phrases

After having completed all of the above steps we are left with 2 sets of vectors: a set of phrase vectors, and a set of concept vectors. Within the context of the i2b2 experiments, each of the concept vectors is also associated with a label. We then assign each phrase a label by

Table 2
The labels assigned to the semantic types and groups.

Label	Semantic Types or Semantic Groups
Problem	Disorders
Treatment	Therapeutic or preventive procedure, Clinical drug
Test	Laboratory procedure, Laboratory or test result, Diagnostic procedure

Table 3
The distribution of labels in our set of concepts.

	# Occurrences
Problem	32,914
Treatment	4206
Test	3855
NP	137,368

calculating the cosine similarity between the phrase vector and all concept vectors.

Formally,

$$\text{sim}(\text{phrase}, \text{concept}) = \frac{\text{phrase} \cdot \text{concept}}{\|\text{phrase}\|_2 \cdot \|\text{concept}\|_2} \quad (8)$$

where phrase and concept are the phrase and concept vectors respectively, and $\|\cdot\|_2$ denotes the euclidean or L2 norm.

$$\text{assign}(\text{phrase}) = \arg \max_{c \in C} \text{sim}(\text{phrase}, c) \quad (9)$$

The assigned concept for a given phrase is then the concept c from the set of concepts C with the largest cosine similarity to the phrase.

Depending on the setting, we then assign the CUI of the concept vector with the highest cosine similarity, or the label assigned to this concept vector, to the phrase.

5. Experiments and results

In this section we present the experimental setups we used to test the efficacy of the model, as well as analyze the obtained results. Experiments 1 and 2 are k NN experiments, designed to reveal the assumptions behind the model. Experiment 3 is a direct comparison to CubNER, and involves assigning labels from the i2b2 label set to extracted noun phrases. Note that only the third experiment involves the concept vectors described in Sections 4.2.2 and 4.2.3. The first two experiments are completely intrinsic, and evaluate the phrase vectors compared to the gold standard labels from the i2b2-2010 corpus.

In all experiments, we evaluate using the F1-score metric, which is the harmonic mean of Precision (P) and Recall (R), which are both calculated using the amount of True Positives (TP), False Positives (FP), and False Negatives (FN). In all our experiments, we use Macro-averaged Precision, Recall, and F1, where applicable.

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

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

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (12)$$

5.1. Experiment 1: k NN on the validation set

Experiment 1 is a k Nearest Neighbors (k NN) experiment on the validation set, intended to test the choices we made during the construction of the model. For all validation experiments, we used k NN classification on the patient notes from the training portion of the i2b2 corpus which came from the Beth Israel Deaconess medical center.

5.1.1. Evaluation methodology

We evaluate our model as follows: for each of the noun phrases extracted from the documents by cTakes [46], we check whether it overlaps with any of the gold standard chunks from the i2b2 corpus. If it does, the label of that specific gold standard chunk gets assigned to the phrase. Any gold standard chunks which do not have any overlap with a noun phrase, or have overlap with more than one noun phrase, are counted as

false negatives. Similarly, any noun phrases which overlap with more than one gold standard chunk are counted as false positives, and removed. Noun phrases which do not overlap with any gold standard chunk are kept, and given the label “np”. These should cluster together with other chunks with the label “np”. This assignment strategy might seem arbitrary, but this is the only arrangement that leads to a constant number of items, regardless of the parser or chunker used.

After having assigned each phrase a label, we can test our system using *k*NN. For each phrase which was assigned a label, we calculate the cosine similarity to all other phrases, and look at the most frequent label within the *k* closest neighbors, and assign that label to the phrase. In all experiments, we set *k* to 1.

Additionally, we also test our model in a “perfect” setting, in which we use the gold standard chunk boundaries as phrase boundaries. This allows us to examine the effect of the chunker on system performance, and shows us an “ideal world” scenario, in which we have a perfect chunker.

We test four different models:

- focus Only includes the focus words, and does not include any context.
- context Only includes context, and leaves out the focus words. This model gives an indication of the system performance on noun phrases without any known words.
- full Includes both focus words and context.
- baseline Uses a Bag of Words (BoW) representation. This model can be directly compared to the **focus** model, and gives an approximation of the effect of the word representations.

The baseline model is constructed directly on the training set by simply counting the words in these documents and assigning the 10,000 most frequent words a feature in a one-hot encoded vector space. An important caveat regarding the baseline is that even if these models perform on par with the regular models, it is still not tractable in practice, because they take a large amount of RAM. In addition, the run time of the baseline model more than doubled.

While the evaluation method we use is supervised, as it requires gold standard data to function, it only serves as a test of the quality of our representations; e.g. if the nearest neighbors of gold standard chunks labeled “problem” are, on average, also labeled “problem”, the clustering was successful. The phrase creation does not involve any labeled data; even though the evaluation is supervised, the system can be fully employed without labeled data, as we show in Section 5.3.

5.1.2. Results

The results of Experiment 1 are shown in Table 4. As we can see, the **context** model, which only uses the context vectors, is clearly the worst-performing model. This is unsurprising, as contexts do not necessarily provide us with a clue towards the meaning. Consider, for example, the fragment “Patient had X”. Without any information about *X*, this context can accommodate a test (“an X-ray”), a treatment (“some aspirin”), or a problem (“a heart attack”). While in many cases, context is bound to be more informative, using context as the sole predictor for the class does not work. Even so, this model provides us with an approximation of how well we can cluster concepts in the case where we do not have word representations for the phrases we are trying to

Table 4

F-scores per class per system in Experiment 1 in the setting where we align the gold standard chunks with the parsed chunks. Bold numbers indicate the best performing system for that class.

	Problem	Treatment	Test	NP	Average
Focus	.77	.78	.80	.86	.80
Context	.32	.35	.31	.87	.43
Full	.76	.76	.80	.90	.80
Baseline	.74	.73	.75	.79	.78

Table 5

F-scores per class per system in Experiment 1 in the perfect setting, where we use the gold standard chunks phrases. Bold numbers indicate the best performing metric for that class.

	Problem	Treatment	Test	Average
Focus	.95	.93	.94	.94
Context	.69	.71	.67	.68
Full	.95	.94	.95	.94
Baseline	.90	.89	.89	.89

extract. (see Table 5).

Comparing the **full** to the **focus** model allows us to gauge the influence context has if we do have access to phrase information. Surprisingly, the addition of context to the phrase itself only slightly improves performance in the perfect setting, and leads to slightly decreased performance in the parsed setting. This shows that a naive, sequential, view of context is probably not sufficiently informative.

Finally, it is surprising that the baseline model does so well, given that it does not really have access to any semantic information whatsoever. In any case, it shows that the targets in the gold standard of the i2b2-2010 corpus have a lot of lexical overlap, as this is the only cue the baseline model can rely on.

5.1.3. The effect of *k*

As mentioned above, the results presented in the previous section were all obtained by using *k*NN with *k* = 1. Nevertheless, analyzing the decrease in F-score for increasing values of *k* can reveal how robust our models are to noise. Figs. 3 and 4 show the fluctuation for increasing values of *k* in the normal setting and perfect setting, respectively.

These figures show that the baseline, performing nearly on par with the other systems for *k* = 1, rapidly decreases in performance. This is in sharp contrast to our systems, which maintain their performance for most values of *k*. In the perfect setting, this results in the baseline system dropping below the performance of the context system when *k* is higher than 24.

5.2. Experiment 2: transferring between hospitals

One of the goals of unsupervised representation learning is the creation of representations which can be reused in different contexts without additional training. We test how well our representations can be transferred between hospitals by creating representations on the documents from the Partners Health clinics, and using these to classify phrases from the Beth Israel Deaconess clinic. This experiment thus provides us with additional evidence regarding the efficacy of our method, as well as providing us with a test of transfer; that is, how well our representations survive the transposition from one clinic to another.

Note that, in the case of the i2b2-2010 corpus, the transfer between the clinics is bound to be relatively minor; both corpora ultimately contain patient notes from the ICU of American clinics. In an ideal setting, we would be able to transfer our representations across a wide range of disciplines and specialties.

We evaluate the same five models as in Experiment 1, using the same evaluation strategy, with the exception that we do not evaluate the quality of the parsed noun phrases, and hence do not provide scores for “perfect” chunking.

5.2.1. Results

The results of Experiment 2 are summarized in Table 6. As in Experiment 1, the **focus** and **full** models outperform all others. An interesting pattern of performance can be seen in the context model, which notably suffers more of a performance hit than the other models in the NP category, but experiences less severe drops for the i2b2-2010 labels. The baseline model, on the other hand, experiences more severe drops in performance, which shows that its generalization performance

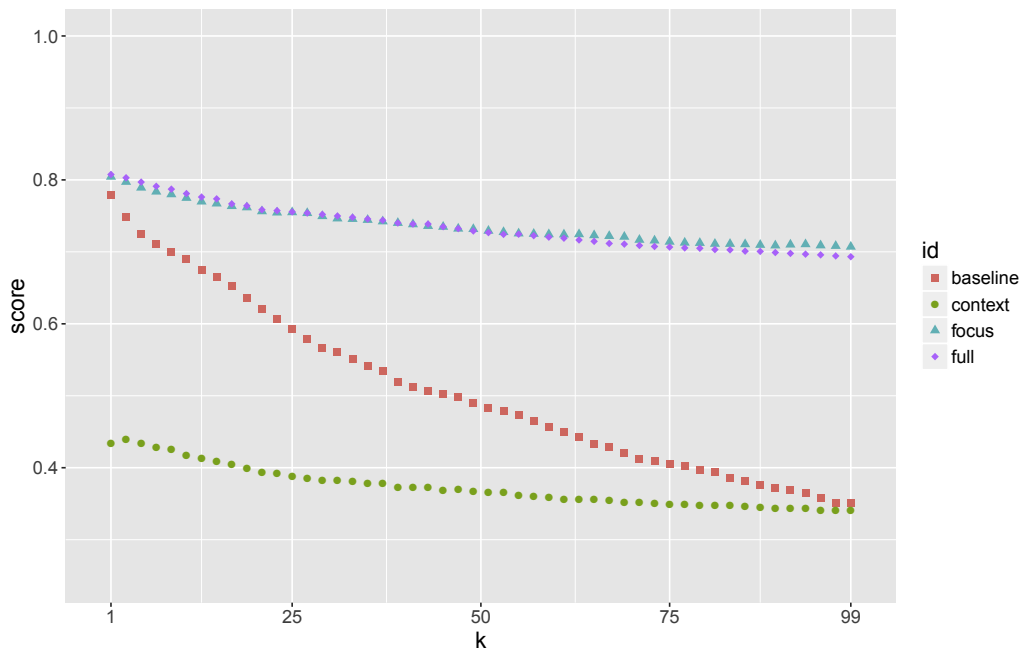


Fig. 3. Average F-scores for increasing values of k in the normal setting.

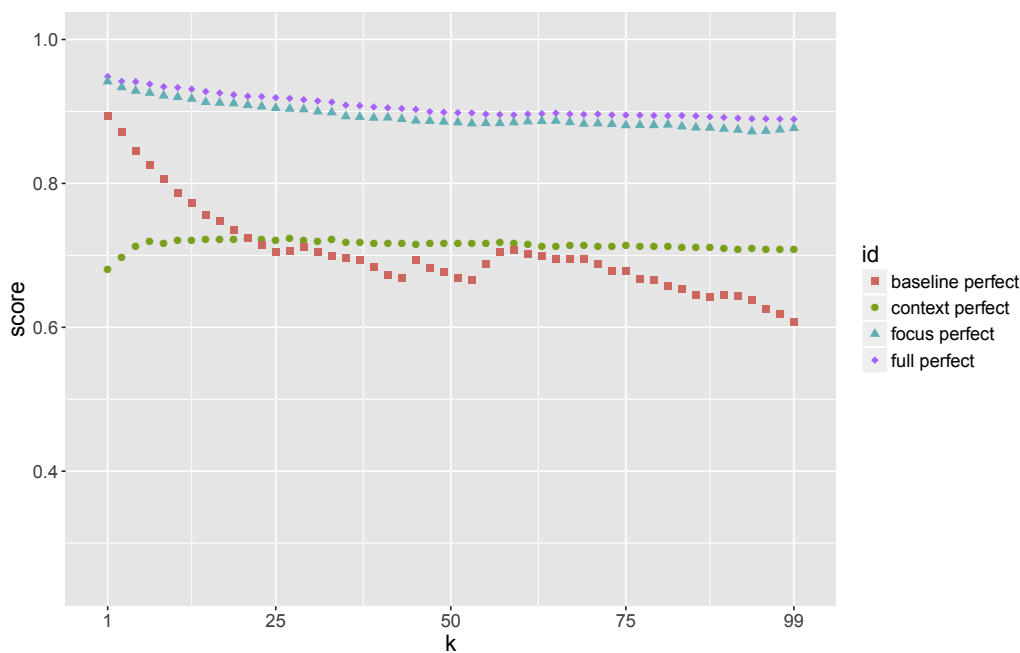


Fig. 4. Average F-scores for increasing values of k in the perfect setting.

Table 6

The F-score per class for each system in Experiment 2. The scores between braces indicate the drop in performance when transferring between hospitals. Bold numbers indicate the best performing system for that class.

	Problem	Treatment	Test	NP
Focus	.70 (-.07)	.64 (-.14)	.70 (-.10)	.87 (+.01)
Context	.25 (-.10)	.31 (-.04)	.22 (-.09)	.71 (-.16)
Full	.68 (-.09)	.64 (-.12)	.69 (-.11)	.91 (+.01)
Baseline	.63 (-.11)	.57 (-.16)	.56 (-.19)	.76 (-.03)

is limited compared to the other models. Finally, this experiment does show that every model experiences at least some kind of negative effect of transfer, which was expected.

5.3. Experiment 3: assigning concepts

In this experiment we directly compare our system to CubNER. We evaluated both our system and CubNER on the test portion of the corpus. As in Experiments 1 and 2, we used the Beth Israel Deaconess portion of the training set as a development set, and thus did not use the test set in any of our development or tuning.

We ran CubNER using the source code provided by the original authors,³ using the parameters mentioned in the paper and the 2015 version of the UMLS. We evaluate all models using Precision, Recall, and macro-averaged F1-score. Following i2b2-2010 guidelines, we use

³ <http://people.dbmi.columbia.edu/szhang/ner.html>.

Table 7

Precision, Recall, and F1-score results of all systems in Experiment 3, as well as the score of the best supervised system on the same dataset. Bold numbers indicate the best performing system for that metric.

	Exact			Inexact		
	P	R	F	P	R	F
Focus	.438	.252	.320	.617	.355	.451
Full	.393	.254	.308	.561	.363	.441
Baseline	.336	.206	.256	.469	.287	.357
CubNER	.285	.241	.261	.491	.414	.449
de Bruijn et al. [50]	.836	.866	.852	.927	.927	.924

two different evaluation settings. In the **exact** setting, a chunk is counted as being correct if the start and end indices match exactly with a gold standard chunk which has the same label. In the **inexact** setting, an extracted chunk only needs to have some overlap with a gold standard chunk with the same label to be counted as correct, with the added constraint that each predicted or gold chunk can only be counted as correct once. Thus, if a predicted chunk overlaps with two gold standard chunks, both of which have the label of the predicted chunk, this will only count as a single True Positive, and the second gold standard chunk will be counted as a false negative.

The exact setting is generally a more realistic setting for NER, but can also be considered too strict. For example, the inclusion or exclusion of determiners can cause an extraction to be counted as incorrect, even though this has little impact on the extracted concepts; “The left ventricle” likely denotes the same concept as “left ventricle”.

We use the CoNLL-2002 shared task evaluation scripts [5] to evaluate the exact setting, while using our own code to evaluate the inexact setting.

5.3.1. Results

The results can be found in Table 7. We see that our model outperforms CubNER in both the exact and inexact settings, although the margin in the Inexact setting is rather small. In the exact setting, we see that our models in general have high precision compared to recall, but still slightly outperform CubNER in terms of recall. In the inexact setting, the story is a bit more nuanced, and CubNER obtains a high recall score, causing it to outperform the **full** model. Such a big difference in performance between the exact and inexact setting might be a reflection of the quality of the parsing performed in CubNER instead of the quality of the system itself.

We also compare our system to the best-performing system on the i2b2 shared task by Bruijn et al. [50]. As we can see, the best-performing supervised system outperforms the unsupervised systems by a large margin.

To see the influence of the chunker, we again evaluated our systems in the perfect setting, in which we use the gold standard chunks as targets. These results are shown in Table 8. Here we see that the addition of context again helps, and that the difference between the system and the baseline gets bigger. This shows that advances in clinical parsing and chunking could greatly influence the accuracy of our system in NER.

In contrast to experiments 1 and 2, we see that the performance of the baseline drops sharply in comparison to the other models. That is, while the baseline was good enough to link phrases with similar phrases in the kNN experiments, it clearly suffers from not having access to semantic information when linking to concepts.

Note that the results presented for CubNER here differ from those presented by Zhang and Elhadad [8] in their original paper, as they do not report scores using the test set. Even so, the scores we found when testing on the Beth Israel Deaconess portion of the corpus were also different; but only slightly so for the exact evaluation.

Table 8

Precision, Recall, and F1-score results of all systems in Experiment 3 if we assume perfect chunking. Bold numbers indicate the best performing system for that metric.

	Perfect		
	P	R	F
Focus	.882	.367	.519
Full	.870	.369	.519
Baseline	.811	.309	.449
CubNER	–	–	–

One distinct possibility for this difference in performance is that the UMLS version causes a difference in performance; in the original paper the 2012AB version of UMLS was used, while we use the 2015AB version in all our experiments, including those with CubNER. However, the authors themselves posit that CubNER is agnostic with regard to UMLS versions. So, while the difference in score might be a reflection of a bias in the model towards a specific version of UMLS, this does not invalidate the scores of the model in this setting.

A larger difference was observed in the inexact setting; the original paper reports F-scores around .50, which is .05 higher than the scores we obtained. As such, this is highly likely to be a difference in calculation of the inexact score rather than a true qualitative difference between runs. A complicating factor is that we do not know which scripts they used to evaluate, and what the exact evaluation criteria for inexact overlap were. In the end, we think that we were able to provide a fair comparison between both models, despite the difference in reported scores.

6. Discussion

6.1. Analyzing the chunker

In this section, we briefly analyze the output of the cTakes chunker. The results of Experiment 1 already showed that moving from the perfect to the parsed setting already costs us about more than 10% points in F-score overall. It is therefore trivial to see that the phrases extracted using the chunker do not match the i2b2-2010 gold standard phrase boundaries perfectly; if they did match perfectly, we would not experience any decrease in performance.

We compare the gold standard chunk boundaries by treating the phrase boundaries extracted by the chunker as predictions in a sequence tagging task. The results of this analysis are shown in Table 9. As the results show, using the phrases extracted by cTakes lead to high recall scores, especially in the inexact setting. Read in another way, the table shows that about 60% of extracted phrases exactly overlap with a gold standard chunk, while about 96% of phrases have partial overlap with a gold standard chunk. Together with the poor precision score in both the exact and inexact setting, we can infer that the chunker extracts a lot of superfluous phrases. Therefore, we conclude that almost all concepts are expressed as (part of) a noun phrase, but that there are a lot of noun phrases which do not contain clinical concepts.

Given the large difference between the exact and inexact settings in this evaluation, we also perform an analysis of the length difference between extracted phrases and gold standard chunks. This shows that of the 2529 phrases which had inexact overlap with a gold standard

Table 9

The results of a standard sequence evaluation when comparing the gold standard chunks versus the phrases extracted using cTakes.

	P	R	F
Exact	.218	.605	.321
Inexact	.350	.968	.514

Table 10

The top 10 superfluous words in noun phrases extracted by cTakes, as compared to the gold standard.

Word	-	No	And	2.	,	3.	am	or	status	neg
#	472	370	140	120	106	88	57	56	41	40

chunk, 1628 phrases (64%) only had a one word length difference, with a mean difference of 1.77. This indicates that the noun phrases largely overlap with the gold standard chunks.

We also perform an analysis of the words in extracted phrases that do not occur in the gold standard chunks. The 10 most common words that occur in phrases without occurring in their gold standard chunks are listed in Table 10. This shows several distinct patterns, including the inclusion of dashes and numerals, which indicates that the cTakes chunker includes items which are typically associated with the start of lists as part of noun phrases. Also of note is the frequent inclusion of “neg” and “no”, which indicates that negation is processed differently by cTakes than expected in the i2b2-2010 annotation guidelines. Given that none of these immediately impact the semantic content of a phrase, e.g. “no hypertension” still contains the concept hypertension, albeit in negated form, we conclude that noun phrase chunking is a good fit for the task of NER on this specific corpus.

6.2. Is context necessary?

As noted above, the addition of context seems to help very little in all our experiments. In Experiments 1 and 2, the models that included both Focus words and Context increased the F-score very slightly, while in Experiment 3, the Recall of the model that included both Focus words and Context was also slightly raised.

This is contrary to our expectation; we expected the addition of a contextually sensitive representation to significantly improve the overall accuracy of the NER system.

As discussed in the beginning of the paper, previous work on WSD showed that context by itself is enough to achieve good performance on a biomedical WSD dataset [9]. Given the low score of the **context** system in this paper, we see that the same does not hold for concept extraction. Of course, concept extraction is a different task than WSD; in a WSD setting, the number of available candidates is usually drastically reduced to two or three alternatives, and sequence extraction does not need to be performed. Similar effects have been observed in clinical spelling correction, where the benefits of context-sensitive disambiguation rapidly decline when the number of possible alternatives increases [51].

An added complication is that, in some sense, adding context is necessary; many concepts are not distinguishable through their surface forms, and the only way to distinguish these words is exactly through the context in which they occur. As such, the results obtained in this paper require extra investigation; it is clear that adding contextual information is necessary to be able to distinguish concepts with the same lexical form from each other, but context does not seem to help much.

Thus, an interesting open question is whether more syntactically informed notions of context could be helpful here. The notion of context used in this paper is extremely simple, and completely ignores the inherently hierarchical nature of language.

7. Conclusion and future work

In this paper we presented a simple method that utilizes pre-trained word embeddings and simple semantic composition to create concept representations. These representations can be linked to noun phrases which are extracted with a chunker, thereby creating an unsupervised system for the extraction of concepts. Our model, although faring notably worse than supervised models on the same dataset, outperforms the only other unsupervised model on the same corpus.

As far as future work goes, we clearly saw that the addition of context to the semantic composition only helped very little. Interesting future work would be to research new ways of compositional semantics that do not necessarily involve conditioning on a supervised target. An easy way to add notions of syntax is to use parse trees, for example those produced by cTakes, and to use syntactic neighbors for composition, as in Socher et al. [22]. Another avenue that could prove fruitful is the use of tree-structured auto-encoders [22] and similar models [52,53], trained on large corpora such as the MIMIC-III corpus.

Finally, the current experiments were carried out using a subset of the available CUIs from the UMLS Metathesaurus, mostly due to performance reasons. While results are promising, future work should focus on evaluating the impact of using such a subset, and on whether CUIs can be aggregated in meaningful ways to form representations of concept hierarchies.

All the code for running the system, preprocessing the data, and visualizing the results is open-source, and can be found at: www.github.com/clips/conch

Conflict of interest

The authors declare no conflict of interest.

Acknowledgments

The first author is supported by a PhD scholarship from the FWO Research Foundation - Flanders. Part of this research was carried out in the framework of the Accumulate IWT SBO project, funded by the government agency for Innovation by Science and Technology (IWT). We would like to thank Pieter Fivez for providing us with references on clinical terminology. Deidentified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

References

- [1] S. Šuster, S. Tulkens, W. Daelemans, A short review of ethical challenges in clinical natural language processing, *EACL* 2017, 2017, p. 80.
- [2] D.L. Mowery, B.R. South, L. Christensen, J. Leng, L.-M. Peltonen, S. Salanterä, H. Suominen, D. Martinez, S. Velupillai, N. Elhadad, et al., Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: share/clef health challenge 2013, task 2, *J. Biomed. Semant.* 7 (2016) 43.
- [3] Y. Kim, E. Riloff, J.F. Hurdle, A study of concept extraction across different types of clinical notes, *AMIA Annual Symposium Proceedings*, vol. 2015, American Medical Informatics Association, 2015, p. 737.
- [4] Ö. Uzuner, B.R. South, S. Shen, S.L. DuVall, 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text, *J. Am. Med. Inform. Assoc.* 18 (2011) 552–556.
- [5] E.F. Tjong Kim Sang, Introduction to the conll-2002 shared task: language-independent named entity recognition, *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002, pp. 155–158.
- [6] E.F. Tjong Kim Sang, F. De Meulder, Introduction to the conll-2003 shared task: language-independent named entity recognition, *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, Association for Computational Linguistics, 2003, pp. 142–147.
- [7] Y. Niu, G. Hirst, G. McArthur, P. Rodriguez-Gianolli, Answering clinical questions with role identification, *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, Association for Computational Linguistics, 2003, pp. 73–80.
- [8] S. Zhang, N. Elhadad, Unsupervised biomedical named entity recognition: experiments with clinical and biological texts, *J. Biomed. Inform.* 46 (2013) 1088–1098.
- [9] S. Tulkens, S. Šuster, W. Daelemans, Using distributed representations to disambiguate biomedical and clinical concepts, *ACL* 2016, vol. 320, 2016, p. 77.
- [10] D.A. Lindberg, B.L. Humphreys, A.T. McCray, et al., The unified medical language system, *IMIA Yearbook* (1993) 41–51.
- [11] Z.S. Harris, *Distributional structure*, *Word* 10 (1954) 146–162.
- [12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, 2013.
- [13] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors, *ACL* (1), 2014, pp. 238–247.
- [14] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations

- of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
- [15] S. Henry, C. Cuffy, B.T. McInnes, Vector representations of multi-word terms for semantic relatedness, *J. Biomed. Inform.* 77 (2018) 111–119.
- [16] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, Y. Goldberg, Fine-grained analysis of sentence embeddings using auxiliary prediction tasks, *arXiv preprint arXiv:1608.04207*, 2016.
- [17] G. Frege, Sense and reference, *Philos. Rev.* 57 (1948) 209–230.
- [18] D.A. Titone, C.M. Connine, On the compositional and noncompositional nature of idiomatic expressions, *J. Pragmat.* 31 (1999) 1655–1674.
- [19] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
- [20] O. Irsoy, C. Cardie, Deep recursive neural networks for compositionality in language, *Advances in Neural Information Processing Systems*, 2014, pp. 2096–2104.
- [21] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, *arXiv preprint arXiv:1503.00075*, 2015.
- [22] R. Socher, E.H. Huang, J. Pennin, C.D. Manning, A.Y. Ng, Dynamic pooling and unfolding recursive autoencoders for paraphrase detection, *Advances in Neural Information Processing Systems*, 2011, pp. 801–809.
- [23] M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III, Deep unordered composition rivals syntactic methods for text classification, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1681–1691.
- [24] F. Hill, K. Cho, A. Korhonen, Y. Bengio, Learning to understand phrases by embedding the dictionary, *arXiv preprint arXiv:1504.00548*, 2015.
- [25] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1735–1780.
- [26] J.L. Elman, Finding structure in time, *Cognit. Sci.* 14 (1990) 179–211.
- [27] M.I. Jordan, Serial order: a parallel distributed processing approach, *Advances in Psychology*, vol. 121, Elsevier, 1997, pp. 471–495.
- [28] L. De Vine, G. Zuccon, B. Koopman, L. Sitbon, P. Bruza, Medical semantic similarity with a neural language model, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ACM, 2014, pp. 1819–1822.
- [29] Y. Choi, C.Y.-I. Chiu, D. Sontag, Learning low-dimensional representations of medical concepts, *AMIA Summits Transl. Sci. Proc.* 2016 (2016) 41.
- [30] S.G. Finlayson, P. LePendou, N.H. Shah, Building the graph of medicine from millions of clinical narratives, *Scient. Data* 1 (2014) 140032.
- [31] A.L. Beam, B. Kompa, I. Fried, N.P. Palmer, X. Shi, T. Cai, I.S. Kohane, Clinical concept embeddings learned from massive sources of medical data, *arXiv preprint arXiv:1804.01486*, 2018.
- [32] A.B. Abacha, P. Zweigenbaum, Medical entity recognition: a comparison of semantic and statistical methods, *Proceedings of BioNLP 2011 Workshop*, Association for Computational Linguistics, 2011, pp. 56–64.
- [33] R. Chalapathy, E.Z. Borzeshi, M. Piccardi, Bidirectional lstm-crf for clinical concept extraction, *arXiv preprint arXiv:1611.08373*, 2016.
- [34] M. Habibi, L. Weber, M. Neves, D.L. Wiegandt, U. Leser, Deep learning with word embeddings improves biomedical named entity recognition, *Bioinformatics* 33 (2017) i37–i48.
- [35] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, *arXiv preprint arXiv:1603.01360*, 2016.
- [36] I.J. Unanue, E.Z. Borzeshi, M. Piccardi, Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition, *J. Biomed. Inform.* 76 (2017) 102–109.
- [37] S. Pradhan, N. Elhadad, W. Chapman, S. Manandhar, G. Savova, Semeval-2014 task 7: analysis of clinical text, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 54–62.
- [38] N. Elhadad, S. Pradhan, S. Gorman, S. Manandhar, W. Chapman, G. Savova, Semeval-2015 task 14: analysis of clinical text, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 303–310.
- [39] A.E. Johnson, T.J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, Mimic-iii, a freely accessible critical care database, *Scient. Data* 3 (2016).
- [40] M. Elsner, E. Charniak, M. Johnson, Structured generative models for unsupervised named-entity clustering, *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2009, pp. 164–172.
- [41] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, A. Yates, Unsupervised named-entity extraction from the web: an experimental study, *Artif. Intell.* 165 (2005) 91–134.
- [42] M. Collins, Y. Singer, Unsupervised models for named entity classification, 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.
- [43] R. Munro, C.D. Manning, Accurate unsupervised joint named-entity extraction from unaligned parallel text, *Proceedings of the 4th Named Entity Workshop*, Association for Computational Linguistics, 2012, pp. 21–29.
- [44] L.A. Ramshaw, M.P. Marcus, Text chunking using transformation-based learning, *Natural Language Processing Using Very Large Corpora*, Springer, 1999, pp. 157–176.
- [45] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, *arXiv preprint arXiv:1607.04606*, 2016.
- [46] G.K. Savova, J.J. Masanz, P.V. Ogren, J. Zheng, S. Sohn, K.C. Kipper-Schuler, C.G. Chute, Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications, *J. Am. Med. Inform. Assoc.* 17 (2010) 507–513.
- [47] W. Blacoe, M. Lapata, A comparison of vector-based representations for semantic composition, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 2012, pp. 546–556.
- [48] J. Mitchell, M. Lapata, Vector-based models of semantic composition, *ACL*, 2008, pp. 236–244.
- [49] F. Galkó, C. Eickhoff, Biomedical question answering via weighted neural network passage retrieval, *arXiv preprint arXiv:1801.02832*, 2018.
- [50] B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, X. Zhu, Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010, *J. Am. Med. Inform. Assoc.* 18 (2011) 557–562.
- [51] P. Fivez, S. S?uster, W. Daelemans, Unsupervised context-sensitive spelling correction of english and dutch clinical free-text with word and character n-gram embeddings, *Comput. Linguist. Netherlands J.* 7 (2017) 39–52.
- [52] S.R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C.D. Manning, C. Potts, A fast unified model for parsing and sentence understanding, *arXiv preprint arXiv:1603.06021*, 2016.
- [53] W. Chung, S.R. Bowman, The lifted matrix-space model for semantic composition, *arXiv preprint arXiv:1711.03602*, 2017.