Implicit Entity Recognition in Clinical Documents

Sujan Perera[*], Pablo Mendes[†], Amit Sheth[*], Krishnaprasad Thirunarayan[*], Adarsh Alex[*], Christopher Heid[¶], Greg Mott[¶]

[*]Kno.e.sis Center, Wright State University, Dayton, OH, USA
[†]IBM Research, San Jose, CA, USA
[¶]Boonshoft School of Medicine, Wright State University, Dayton, OH, USA
sujan@knoesis.org, pnmendes@us.ibm.com, amit@knoesis.org, tkprasad@knoesis.org
adarsh@knoesis.org, caheid2@gmail.com, mott11@wright.edu

Abstract

With the increasing automation of health care information processing, it has become crucial to extract meaningful information from textual notes in electronic medical records. One of the key challenges is to extract and normalize entity mentions. State-of-the-art approaches have focused on the recognition of entities that are explicitly mentioned in a sentence. However, clinical documents often contain phrases that indicate the entities but do not contain their names. We term those *implicit entity mentions* and introduce the problem of implicit entity recognition (IER) in clinical documents. We propose a solution to IER that leverages entity definitions from a knowledge base to create entity models, projects sentences to the entity models and identifies implicit entity mentions by evaluating semantic similarity between sentences and entity models. The evaluation with 857 sentences selected for 8 different entities shows that our algorithm outperforms the most closely related unsupervised solution. The similarity value calculated by our algorithm proved to be an effective feature in a supervised learning setting, helping it to improve over the baselines, and achieving F1 scores of .81 and .73 for different classes of implicit mentions. Our gold standard annotations are made available to encourage further research in the area of IER.

1 Introduction

Consider the following sentence, extracted from a clinical document: *"Patient has shortness of breath with reaccumulation of fluid in extremities."* It states that the patient has 'shortness of breath' and 'edema'. The former is explicitly mentioned, while the latter is implied by the semantics of the phrase *'reaccumulation of fluid in extremities'*. We term such occurrences *implicit entity mentions*.

While implicit entity mentions are common in many domains, resolving them is particularly valuable in the clinical domain. Clinical documents are rich in information content that plays a central role in understanding patients' health status and improving the quality of the delivered services. It is a common practice to employ computer assisted coding (CAC) solutions to assist expert "coders" in determining the unique identifier (e.g., ICD9 or ICD10) for each medical condition or combination of conditions. These identifiers are important to unambiguously represent the medical conditions, to prepare the post-discharge plan, and to perform secondary data analysis tasks. A human coder reading the sentence *'Patient has shortness of breath with reaccumulation of fluid in extremities'* would generate the corresponding codes for entities 'shortness of breath' and 'edema'. However, the solutions developed to perform entity recognition in clinical documents (Aronson, 2006) (Friedman et al., 1994) (Savova et al., 2010) (Friedman et al., 2004) (Fu and Ananiadou, 2014) (Pradhan et al., 2015) do not recognize the presence of entity 'edema' in this sentence.

Implicit entity mentions are a common occurrence in clinical documents as they are often typed during a patient visit in a way that is natural in spoken language and meant for consumption by the professionals with similar backgrounds. An analysis with 300 documents in our corpus showed that 35% of the 'edema' mentions and 40% of the 'shortness of breath' mentions are implicit.

Recognizing implicit mentions is particularly

*Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 228–238,
Denver, Colorado, June 4–5, 2015.

challenging since, besides the fact that they lack the entity name, they can be embedded with negations. For example, the semantics of the sentence *'The patients' respiration become unlabored'* implies that the patient does not have 'shortness of breath'. Identification of the negated mentions of entities in clinical documents is crucial as they provide valuable insights into the patients' health status.

We propose an unsupervised solution to the IER problem that leverages knowledge embedded in entity definitions obtained for each entity from the Unified Medical Language System (UMLS) (Bodenreider, 2004). UMLS provides a standard vocabulary for the clinical domain. Our solution: a) Creates an entity model from these definitions, b) Identifies the sentences in input text that may contain implicit entity mentions, c) Projects these sentences onto our entity model, and d) Classifies the sentences to distinguish between those containing implicit entity mentions or negated implicit mentions, by calculating the semantic similarity between the entity model and the projected sentences.

The contributions of this work are as follows:

1. We introduce the problem of implicit entity recognition (IER) in clinical documents.

2. We propose an unsupervised solution to IER that outperforms the most relevant unsupervised baseline and improves the results of a supervised baseline.

3. We create a gold standard corpus annotated for IER in the clinical domain and make it available to encourage research in this area.

2 Related Work

To the best of our knowledge, this is the first work to address the problem of Implicit Entity Recognition (IER) in clinical documents. However, there is a large body of research that is relevant to the problem, including Named Entity Recognition (NER), Entity Linking (EL), Coreference Resolution, Paraphrase Recognition, and Textual Entailment Recognition.

Much like IER, both NER and EL have the objective of binding a natural language expression to a semantic identifier. However, related work in NER and EL expect the proper name (explicit mention) of entities and assume the presence of noun phrases (Collins and Singer, 1999) (Bunescu and Pasca, 2006). The solutions developed for NER leverage regularities on morphological and syntactical features that are unlikely to hold in the case of IER. The most successful NER approaches use word-level features (such as capitalization, prefixes/suffixes, and punctuation), list lookup features (such as gazetteers, lexicons, or dictionaries), as well as corpus-level features (such as multiple occurrences, syntax, and frequency) (Nadeau and Sekine, 2007) that are not exhibited by the phrases with implicit entity mentions.

Many approaches couple NER with a follow up EL step (Hachey et al., 2013) in order to assign an unique entity identifiers to mentions. Therefore, the inadequacy of NER techniques will limit the capability of recognizing implicit entity mentions by a solution developed for EL. Moreover, state-of-the-art EL approaches include a 'candidate mapping' step that uses entity names to narrow down the space of possible entity identifiers, which is also a limiting factor in the IER case. Finally, neither NER nor EL deal with the negated mentions of entities.

Coreference resolution (CR) focuses on grouping multiple mentions of the same entity with different surface forms. The solutions to CR focus on mapping explicit mentions of entity names to other pronouns and noun phrases referring to the same entity (Ng, 2010) (Durrett and Klein, 2013). In IER implicit mentions occur without co-referring corresponding entity. Hence, they must be resolved without dependencies on co-referents.

In contrast to NER, EL, and CR problems and their solutions, IER addresses instances where neither explicit mention of an entity nor noun phrases or any of the above mentioned features are guaranteed to appear in the text but still have a reference to a known entity. Hence, IER solutions require treatment for implied meaning of the phrases beyond its syntactic features.

Since our solution to IER establishes a relationship between entity definitions and the input text, the tasks of paraphrase recognition (Barzilay and Elhadad, 2003) (Dolan et al., 2004) and textual entailment recognition (Giampiccolo et al., 2007) are related to our solution. However, these tasks are fundamentally different in two aspects: 1) Both paraphrase recognition and textual entailment recogni-

tion are defined at the sentence level, whereas text phrases considered for IER can exist as a sentence fragment or span across multiple sentences, and 2) The objective of IER is to find whether a given text phrase has a mention of an *entity*—as opposed to determining whether two sentences are similar or entail one another. However, our solution benefits from the lessons learned from both tasks.

The question answering solutions cope with the questions that describe the characteristics of a concept and expect that concept as the answer. This particular type of questions resembles implicit entity mentions. However, they assume that the questions are referring to some concept and the problem is to uncover which one, whereas the implicit entity mention problem requires us to first check whether a particular sentence/phrase has a mention of an entity at all. Furthermore, question answering systems benefit from the presence of pronouns, nouns, and noun phrases in the questions and the candidate answers to derive helpful syntactic and semantic features (Lally et al., 2012)(Wang, 2006), while phrases with implicit entity mentions may not contain such features.

The existing work on clinical document annotation focused on explicit entity mentions with contiguous phrases (Aronson, 2006) (Savova et al., 2010) (Friedman et al., 2004) (Fu and Ananiadou, 2014). Going one step beyond, the SemEval 2014 task 7 recognized the need for identifying discontiguous mentions of explicit entities (Pradhan et al., 2014). However, the recognition of implicit entities has yet to address by this community.

3 Implicit Entity Recognition (IER) in Clinical Documents

We define the Implicit Entity Recognition (IER) task in clinical documents as: given input text that does not have explicit mentions of target entities, find which target entities are implied (including implied negations) in the input text.

Negation detection is traditionally separated from the entity recognition task because negation indicating terms can be recognized separately from the phrases that contain explicit mention of an entity. In contrast, implicit mention can involve an antonym that fuses the entity indication with negated sense

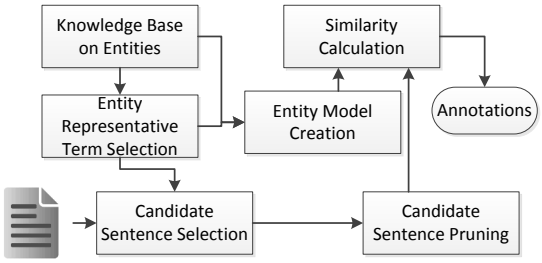

Figure 1: Components of the Proposed Solution

(e.g., *'patient denies shortness of breath'* vs *'patient is breathing comfortably'*). Hence, negation detection is considered as a sub-task of IER.

Typical entity recognition task considers the detection of the boundaries of the phrases with entities (i.e., segmentation) as a sub-task. We consider boundary detection of the implicit entity mentions as an optional step due to two reasons: 1) It is considered an optional step in biomedical entity recognition task (Tsai et al., 2006), and 2) The phrases with implicit entity mentions can be noncontiguous and span multiple sentences. Further, in some cases, even domain experts disagree on the precise phrase boundaries.

We define the IER as a classification task. Given an input text, classify it to one of the three categories: TP_e if the text has a mention of entity e, or $Tneg_e$ if the text has a negated mention of entity e, or TN_e if the entity e is not mentioned at all. As mentioned, the phrases with implicit entity mentions can span to multiple sentences. However, this work will focus only on implicit mentions exist within a sentence. Our unsupervised solution to this classification task: 1) Creates an *entity model* from the entity definitions, 2) Selects *candidate sentences* that may contain implicit entity mentions, 3) Projects the candidate sentences into entity model space, and 4) Calculates the semantic similarity between projected sentences and the *entity model*. Figure 1 shows the components of our solution which are discussed below in detail.

In order to facilitate these sub-tasks, our algorithm introduces the concept of an *entity representative term* for each entity and propose an automatic way to select these terms from entity definitions.

3.1 Entity Representative Term Selection

Entity representative term (ERT) selection finds a term with high *representative power* to an entity and plays an important role in defining it.

The *representative power* of a term t for entity e is defined based on two properties: its dominance among the definitions of entity e, and its ability to discriminate the mentions of entity e from other entities. This is formalized in eq. (1). Consider the entity 'appendicitis' as an example. It is defined as *'acute inflammation of appendix'*. Intuitively, both terms *inflammation* and *appendix* are candidates to explain the entity *appendicitis*. However, the term *appendix* has more potential to discriminate the implicit mentions of *appendicitis* than the term *inflammation*, because the term *inflammation* is used to describe many entities. Also, none of the definitions define *appendicitis* without using the term *appendix*; therefore, *appendix* is the dominant term, and consequently it has the most representative power for the entity 'appendicitis'.

We used a score inspired by the TF-IDF measure to capture this intuition. The IDF (inverse document frequency) value measures the specificity of a term in the definitions. The TF (term frequency) captures the dominance of a term. Hence the representative power of a term t for entity e (r_t) is defined as,

$$r_t = freq(t, \mathcal{Q}_e) * \log \frac{|E|}{|E_t|} \quad (1)$$

$\mathcal{Q}_e$ is the set of definitions of entity e, E is the set of all entities. $freq(t, \mathcal{Q}_e)$ is the frequency of term t in set $\mathcal{Q}_e$, $|E|$ is the size of the set E (3962 in our corpus), and the denominator $|E_t|$ calculates the number of entities defined using term t. We expand the ERT found for an entity with this technique by adding its synonyms obtained from WordNet.

We can define entity representative terms based on the definition of representative power.

Definition 3.1 (Entity Representative Term)**.** Let $\mathcal{L}_e = \{t_1, t_2, ..., t_n\}$ be the set of terms in a definitions of an entity e. Let $\mathcal{R}_{\mathcal{L}_e} = \{r_{t_1}, r_{t_2}, ..., r_{t_n}\}$ be the representative power calculated for each term t_i in $\mathcal{L}_e$ for e. We select term t_m as the entity representative term of the entity e if its representative power is maximum, i.e., $r_{t_m} \geq r_{t_i}$ for all i where $1 \leq i \leq n$.

3.2 Entity Model Creation

Our algorithm creates an *entity indicator* from a definition of the entity. An entity indicator consists of terms that describe the entity. Consider the definition *'A disorder characterized by an uncomfortable sensation of difficulty breathing'* for 'shortness of breath', for which the selected ERT is *'breathing'*. The terms *uncomfortable*, *sensation*, *difficulty*, and *breathing* collectively describe the entity. Adding other terms in this definition to the entity indicator negatively affects the similarity calculation with the candidate sentences since they are less likely to appear in a candidate sentence. We exploited the neighborhood of the ERT in the definition to create the entity indicator and automatically selected the nouns, verbs, adjectives, and adverbs in the definition within a given window size to the left and to the right of the ERT. We used a window size of four in our experiments.

An entity can have multiple definitions each explaining it using diverse vocabulary. On average, an entity in our corpus had 3 definitions. We create an entity indicator from each definition of the entity, hence an entity has multiple indicators. We call the collection of indicators of an entity as its *entity model*. In other words, an entity model consists of multiple entity indicators that capture diverse and orthogonal ways an entity can be expressed in the text.

3.3 Candidate Sentence Selection

The sentences with ERT in an input text are identified as *candidate sentences* containing implicit mention of the corresponding entity. A sentence may contain multiple ERTs and consequently become a candidate sentence for multiple entities. This step reduces the complexity of the classification task as now a sentence has only a few target entities.

3.4 Candidate Sentence Pruning

In order to evaluate the similarity between any given candidate sentence and the entity model, we perform a projection of candidate sentences onto the same semantic space. We perform this by pruning the terms in candidate sentences that does not participate in forming the segment with implicit entity mentions. Candidate sentences are pruned by fol-

lowing the same steps followed to create the entity indicators from the entity definitions.

3.5 Semantic Similarity Calculation

As the last step, our solution calculates the similarity between the entity model and the pruned candidate sentence. The sentences with implicit entity mentions often use adjectives and adverbs to describe the entity and they may indicate the absence of the entities using antonyms or explicit negations. These two characteristics pose challenges to the applicability of existing text similarity algorithms such as MCS (Mihalcea et al., 2006) and matrixJcn (Fernando and Stevenson, 2008) which are proven to perform well among the unsupervised algorithms in paraphrase identification task (ACLWiki, 2014).

The existing text similarity algorithms largely benefit from the WordNet similarity measures. Most of these measures use the semantics of the hierarchical arrangement of the terms in WordNet. Unfortunately, adjectives and adverbs are not arranged in a hierarchy, and terms with different part of speech (POS) tags cannot be mapped to the same hierarchy. Hence, they are limited in calculating the similarity between terms of these categories. This limitation negatively affects the performance of IER as the entity models and pruned sentences often contain terms from these categories. Consider the following examples:

1. *Her breathing is still uncomfortable*$_{adjective}$.
2. *She is breathing comfortably*$_{adverb}$ *in room air.*
3. *His tip of the appendix was inflamed*$_{verb}$.

The first two examples use an adjective and an adverb to mention the entity 'shortness of breath' implicitly. The third example uses a verb to mention the entity 'appendicitis' implicitly instead of the noun *inflammation* that is used by its definition.

We have developed a text similarity measure overcoming these challenges and weigh the contributions of the words in the entity model to the similarity value based on their representative power.

Handling adjectives, adverbs and words with different POS tags: To get the best out of all WordNet similarity measures, we exploited the relationships between different forms of the terms in WordNet to find the noun form of the terms in the entity models and pruned sentences before calculating the similarity. We found the adjective for an adverb using relationship 'pertainym' and noun for an adjective or a verb using the relationship 'derivationally related form' in WordNet.

Handling negations: Negations are of two types: 1) Negations mentioned with explicit terms such as no, not, and deny, and 2) Negations indicated with antonyms (e.g., 2nd example in above list). We used the NegEx algorithm (Chapman et al., 2001) to address the first type of negations. To address the second type of negations, we exploited the antonym relationships in the WordNet.

The similarity between an entity model and the pruned candidate sentence is calculated by computing the similarities of their terms. The term similarity is computed by forming an ensemble using the standard WordNet similarity measures namely, WUP (Wu and Palmer, 1994), LCH (Leacock and Chodorow, 1998), Resnik (Resnik, 1995), LIN (Lin, 1998), JCN (Jiang and Conrath, 1997), as well as a predict vector-based measure Word2vec (Mikolov et al., 2013) and a morphology-based similarity metric Levenshtein[1] as:

$$sim(t_1, t_2) = max_{m \in M}(sim_m(t_1, t_2)) \quad (2)$$

where t_1 and t_2 are input terms and M is the set of above mentioned similarity measures. This ensemble-based similarity measure exploits orthogonal ways of comparing terms: semantic, statistical, and syntactic. An ensemble-based approach is preferable over picking one of them exclusively since they are complementary in nature, that is, each outperforms the other two in certain scenarios.

The similarity values calculated by WordNet similarity measures in $sim_m(t_1, t_2)$ are normalized to range between 0 and 1.

The similarity of a pruned candidate sentence to the entity model is calculated by calculating its similarity to each entity indicator in the entity model, and picking the maximum value as the final similarity value for the candidate sentence. The similarity between entity indicator e and pruned sentence s, $sim(e, s)$, is calculated by summing the similarities calculated for each term t_e in the entity indicator weighted by its representative power as defined

[1] http://en.wikipedia.org/wiki/Levenshtein_distance

in eq. (1). If t_e is an antonym for any term in s (t_s), it contributes negatively to the overall similarity value, else it contributes in linear portion of the maximum similarity value between t_e and some t_s (eqs. (4) and (5)). The overall similarity value is normalized based on the total representative power of all the terms t_es (eq. (1)) and ranges between -1 and +1.

Note that this formulation weighs the contribution of each term according to its importance in defining the entity. The higher similarity with a term that has higher representative power leads to higher overall similarity value, while the lower similarity with such terms leads to a lower total similarity value. The special treatment for antonyms takes care of the negated mentions of an entity.

$$sim(e, s) = \frac{\sum_{t_e \in e} f(t_e, s) * r_{t_e}}{\sum_{t_e \in e} r_{t_e}} \quad (3)$$

$$f(t_e, s) = \begin{cases} -1 & \alpha(t_e, s) == 0 \\ \max_{t_s \in s} \text{sim}(t_e, t_s) & \text{otherwise} \end{cases} \quad (4)$$

$$\alpha(t_e, s) = \prod_{t_s \in s} \begin{cases} 0 & \text{if } t_e \text{ is an antonym of } t_s \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Finally, the sentences are classified based on a configurable threshold values selected between -1 and +1.

4 Evaluation

We reannotated a sample of the corpus created for SemEval-2014 task 7 (Pradhan et al., 2014) to include implicit mention annotations and measured the performance of our proposed method in classifying entities annotated with *TP* and *Tneg* mentions[2].

4.1 Gold Standard Dataset

The SemEval-2014 task 7 corpus consists of 24,147 de-identified clinical notes. We used this corpus to create a gold standard for IER with the help of three domain experts. The gold standard consists of 857 sentences selected for eight entities. The creation of the gold standard is described below in detail.

Entity	*TP*	*Tneg*	*TN*
Shortness of breath	93	94	29
Edema	115	35	81
Syncope	96	92	24
Cholecystitis	78	36	4
Gastrointestinal gas	18	14	5
Colitis	12	11	0
Cellulitis	8	2	0
Fasciitis	7	3	0

Table 1: Candidate Sentence Statistics

We have annotated the corpus for explicit mentions of the entities using cTAKES (Savova et al., 2010) and ranked the entities based on their frequency. The domain experts on our team then selected a subset of these entities that they judged to be frequently mentioned implicitly in clinical documents. For example, the frequent entity 'shortness of breath' was selected but not 'chest pain' since the former is mentioned implicitly often but not the latter. We used four frequently implicitly mentioned entities as the primary focus of our evaluation. We refer to these as *primary entities* from here on (the first four entities in Table 1). To test the generalizability of our method, as well as to evaluate its robustness when lacking training data, we selected another four entities (the last four entities in Table 1). We then selected a random sample of candidate sentences for each of these entities based on their ERTs and winnowed it down further by manually selecting a subset that exhibits syntactic diversity. Ultimately, our corpus consisted of 120-200 sentences for each primary entity and additional 80 sentences selected from the other four entities.

Each candidate sentence was annotated as TP_e (contains a mention of entity e), $Tneg_e$ (contains a negated mention of entity e), or TN_e (does not contain a mention of entity e). Each sentence was annotated by two domain experts, and we used the third one to break the ties. The Cohens' kappa value for the annotation agreement was 0.58. While the annotators have good agreement on annotating sentences in category *TP*, they agreed less on the categories *Tneg* and *TN*. The latter categories are indeed difficult to distinguish. For example, annotators often argue whether *'patient breathing at a rate of 15-20'*

[2]We do not explicitly report performance on *TN* because our focus is to find sentences that contain entity mentions rather than those devoid of mentions.

means the negation of entity 'shortness of breath' (because that is a normal breathing pattern) or just lacks a mention of the entity. The final annotation label for a sentence is decided based on majority voting. Table 1 shows the statistics of the annotated candidate sentences. The prepared data set is available at `http://knoesis.org/researchers/sujan/data-sets.html`

4.2 Implicit Entity Recognition Performance

Since IER is a novel task, there are no baseline algorithms that can be directly applied such that it would yield a fair comparison with our algorithm. However, we deem some of the related algorithms to have good potential applicability for this task. Therefore, we included two strong algorithms from the closest related work as baseline solutions to the problem.

The first baseline is the well-known text similarity algorithm MCS (Mihalcea et al., 2006). MCS is one of the best performing unsupervised algorithms in paraphrase recognition task (ACLWiki, 2014). It uses an ensemble of statistical and semantic similarity measures, which is a preferable feature for the IER as opposed to one measure used by the matrixJcn (Fernando and Stevenson, 2008). Both MCS and our algorithm classify the candidate sentences based on threshold values selected experimentally.

To include also a supervised baseline, we trained an SVM (Cortes and Vapnik, 1995) one of the state-of-the-art learning algorithms, shown to perform remarkably well in a number of classification tasks. We trained separate SVMs for each primary entity, considering unigrams, bigrams, and trigrams as the features. It has been shown that SVM trained on ngrams performed well on text classification tasks (Pang et al., 2002) (Zhang and Lee, 2003). The SVMs trained with bigrams consistently produced the best results for the 4-fold cross validation. Therefore, our testing phase used the SVMs trained with the bigrams.

Preparation of training and testing datasets: We created training and testing datasets by splitting the dataset annotated for each primary entity as 70% (training) and 30% (testing). The training datasets were used to train the SVM models for each primary entity and to select the threshold values for both MCS and our algorithm.

The classification performance of each algorithm is studied in the *TP* and *Tneg* categories using precision, recall, and F-measure.

The precision (*PP*) and recall (*PR*) for category *TP* at threshold t are defined as:

$$PP_t = \frac{S_{TP}\ with\ sim \geq t}{all\ sentences\ with\ sim \geq t}$$
$$PR_t = \frac{S_{TP}\ with\ sim \geq t}{S_{TP}}$$

Similarly, *NP* and *NR* for *Tneg* are defined as:

$$NP_t = \frac{S_{Tneg}\ with\ sim < t}{all\ sentences\ with\ sim < t}$$
$$NR_t = \frac{S_{Tneg}\ with\ sim < t}{S_{Tneg}}$$

where S_{TP} and S_{Tneg} denote the sentences annotated with *TP* and *Tneg* respectively by domain experts and sim is the calculated similarity value for the pruned sentence.

Selecting threshold value: The threshold values for both MCS and our algorithm are selected based on their classification performance in the training dataset. The MCS algorithm produced the best F1 score for the *TP* category with a threshold value of 0.5, and for the *Tneg* category with a threshold value of 0.9, while our algorithm produced the best F1 for the *TP* category with 0.4 and for the *Tneg* category with 0.3. We examined threshold values that produce best F1 scores by the two algorithms by starting with 10% of the training data and gradually increasing the size of the training data. The threshold values with best F1 scores were stabilized after adding 30% of the training data. Hence, we could select the threshold values with just 50% of the training data.

4.3 Classification Performance

The first experiment evaluates the classification performance of our algorithm, MCS, and SVM.

Method	*PP*	*PR*	*PF1*	*NP*	*NR*	*NF1*
Our	0.66	0.87	0.75	**0.73**	0.73	**0.73**
MCS	0.50	**0.93**	0.65	0.31	**0.76**	0.44
SVM	**0.73**	0.82	**0.77**	0.66	0.67	0.67

Table 2: precision, recall, and F1 values for each algorithm (PF1 and NF1 indicate F1 scores for the *TP* and *Tneg* categories respectively). SVM outperforms our algorithm in the *TP* category, while our algorithm outperforms SVM on the *Tneg* category.

Our algorithm outperforms the other unsupervised solution MCS, but the SVM was able to leverage supervision to outperform our algorithm in the *TP* category in terms of F-measure ($PF1$ on Table 2). For example, the sentence *'he was placed on*

mechanical ventilation shortly after presentation' is annotated as *TP* in the gold standard for the entity 'shortness of breath' since *'mechanical ventilation'* indicates the presence of 'shortness of breath'. This annotation requires domain knowledge that was not present in the entity definitions that we used to build entity models. However, with enough examples, the SVM was able to learn the importance of the bigram *'mechanical ventilation'* and classify it as *TP*.

For the *Tneg* category, however, our algorithm outperforms the SVM ($NF1$ on Table 2). This is due to the explicit treatment for the negated mentions by our algorithm to capture different variations of the negated mentions.

The MCS algorithm underperformed in both categories. We observed that this was mostly due to its limitations described in Section 3.5. The overall classification accuracy—the accuracy of classifying both *TP* and *Tneg* instances—of our algorithm, MCS, and SVM are 0.7, 0.4, and 0.7 respectively.

Method	*PP*	*PR*	*PF1*	*NP*	*NR*	*NF1*
SVM	0.73	0.82	0.77	0.66	0.67	0.67
SVM+MCS	0.73	0.82	0.77	0.66	0.66	0.66
SVM+Our	**0.77**	**0.85**	**0.81**	**0.72**	**0.75**	**0.73**

Table 3: Comparison of SVM results incorporating similarity values calculated by our algorithm and MCS as a feature. Our algorithm complements the SVM in both categories whereas MCS does not contribute to improve the classification.

The second experiment evaluates the impact of including the similarity scores calculated by MCS and our algorithm for each candidate sentence as a feature to the best performing SVM model. Table 3 shows that the inclusion of MCS scores as a feature did not help to improve the SVM results. In fact, it negatively affected the results for the *Tneg* category. Since the MCS showed low precision for the *Tneg* category in the previous experiment (Table 2), it is potentially introducing too much noise that the SVM is not able to linearly separate. However, the similarity value calculated by our algorithm improves the SVM classifiers. It increased the precision and recall values for both the *TP* and *Tneg* categories. This shows that the similarity value calculated by our algorithm can be used as an effective feature for a learning algorithm that is designed to solve the IER problem. The overall classification accuracy of SVM, SVM+MCS, and SVM+Our configurations are 0.7, 0.7, and 0.74 respectively.

We were interested in exploring how much labeled data would be needed for supervised solution to outperform our unsupervised score alone. We analyzed the behavior of the three configurations of the SVM with our unsupervised approach with different training set sizes. Figure 2 shows the F1 values obtained by gradually increasing the size of the training dataset[3], while testing on the same test set. The F1 value of our approach remains constant after 50% training data since it has already decided the threshold values. Figure 2 shows that the SVM trained with bigrams needs 76% of the training dataset to achieve the F1 value achieved by our unsupervised approach in the *TP* category, and it does not achieve the F1 achieved by our algorithm in *Tneg* category (note the crossing points of the line marked with 'X' and line marked with circles).

Figure 2 also shows that the similarity score calculated by our algorithm complements the SVM at each data point. After adding our similarity score to the SVM as a feature, it achieved the F1 achieved by our unsupervised algorithm with 50% of the training data in the *TP* category and with 90% of the training data in the *Tneg* category (note the crossing points of the line marked with 'X' and line marked with '+'). Also, SVM+Our configuration achieved the best F1 value for SVM with just 70% of the training data in the *TP* category and with just 50% of the training data in the *Tneg* category. This shows that our similarity score can be used as an effective feature to reduce manual labeling effort and to improve the supervised learning algorithms to solve the IER problem.

Finally, to evaluate the generalization ability of our algorithm and to demonstrate its usage in situations with a lack of training data, we applied it to a set of 80 sentences selected for four new entities (the last four entities in Table 1). Our algorithm produced the following results for these entities when we classify their sentences with the threshold values selected using the training dataset created for the primary entities.

$$PP = 0.72, PR = 0.77, PF1 = 0.74$$
$$NP = 0.78, NR = 0.83, NF1 = 0.80$$

[3]We draw these graphs considering training dataset size >50% for clarity.

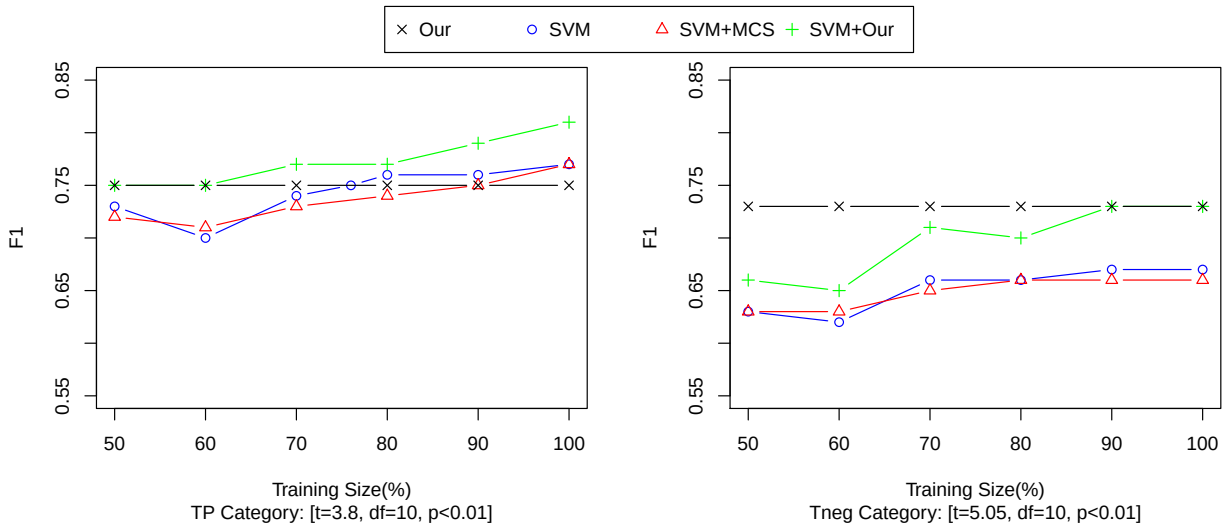

Figure 2: The variation of the F1 value in the *TP* (left) and *Tneg* (right) categories with varying sizes of training datasets. These graphs show that the SVM trained with bigrams needs 76% of the training data to achieve the F1 score of our unsupervised method in the *TP* category while it does not achieve the F1 score of our algorithm in the *Tneg* category. This also shows that the similarity value calculated by our algorithm complements the SVM trained with bigrams at each data point and helps it to beat or perform on par with our algorithm. The paired T-test values calculated for SVM and SVM+Our configurations show that this is not a random behavior (t- T-test value, df- degree of freedom, p- probability value).

Although negation detection with NegEx is not a contribution of our work, our algorithm enables its application to IER. This is not possible for MCS. NegEx requires two inputs: 1) The sentence, and 2) The term being considered for possible negation. MCS does not detect the key term in the sentence, hence it is not possible to apply NegEx with MCS. However, our algorithm starts with identifying the ERT which is considered for possible negation.

5 Limitations

The candidate sentence selection based on the ERT can be seen as a limitation of our approach since it does not select sentences with implicit entity mentions that do not use the selected ERT. However, we do not expect this limitation to have a major impact. We asked our domain experts to come up with sentences that contain implicit mentions of the entity 'shortness of breath' without using its ERT *'breathing'* or its synonyms (*'respiration'* and *'ventilation'*). According to them, the sentences *'the patient had **low oxygen saturation**'*, *'the patient was **gasping for air**'*, and *'patient was **air hunger**'* are such sentences (the emphasis indicates the phrases that imply 'shortness of breath'). However, we found only 113 occurrences of these phrases as opposed to 8990 occurrences of its ERTs in our corpus.

6 Conclusion and Future Work

We defined the problem of *implicit entity recognition* in clinical documents and proposed an unsupervised solution that recognizes the implicit mentions of entities using a model built from their definitions in a knowledge base. We showed that our algorithm outperforms the most relevant unsupervised method and it can be used as an effective feature for a supervised learning solution based on an SVM. The ability to capture the diverse ways in which an entity can be implicitly mentioned by exploiting their definitions with special treatment for two types of negations are the main strengths of our method.

In the future, we will explore the ability to detect the boundary of the phrases with implicit mentions, capture the sentences with implicit mentions without selected ERT, and investigate more intensive exploitation of domain knowledge for IER.

7 Acknowledgement

We acknowledge the medical students Logan Markins, Kara Joseph, and Robert Beaulieu of Wright State University for their assistance in creating the gold-standard corpus.

References

ACLWiki. 2014. Paraphrase identification (state of the art). http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art). [Online; accessed 19-Dec-2014].

Alan R Aronson. 2006. Metamap: Mapping text to the umls metathesaurus. *Bethesda, MD: NLM, NIH, DHHS*.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 25–32. Association for Computational Linguistics.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

Razvan C Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16.

Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110. Citeseer.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*, pages 1971–1982.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, pages 45–52. Citeseer.

Carol Friedman, Philip O Alderson, John HM Austin, James J Cimino, and Stephen B Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174.

Carol Friedman, Lyudmila Shagina, Yves Lussier, and George Hripcsak. 2004. Automated encoding of clinical documents based on natural language processing. *Journal of the American Medical Informatics Association*, 11(5):392–402.

Xiao Fu and Sophia Ananiadou. 2014. Improving the extraction of clinical concepts from clinical records. *Proceedings of BioTxtM14*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with wikipedia. *Artif. Intell.*, 194:130–150, January.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Adam Lally, John M Prager, Michael C McCord, BK Boguraev, Siddharth Patwardhan, James Fan, Paul Fodor, and Jennifer Chu-Carroll. 2012. Question analysis: How watson reads a clue. *IBM Journal of Research and Development*, 56(3.4):2–1.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1396–1411, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Sameer Pradhan, Noémie Elhadad, Wendy Chapman, Suresh Manandhar, and Guergana Savova. 2014. Semeval-2014 task 7: Analysis of clinical text. *SemEval 2014*, 199(99):54.

Sameer Pradhan, Noémie Elhadad, Brett R South, David Martinez, Lee Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana Savova. 2015. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *Journal of the American Medical Informatics Association*, 22(1):143–154.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. 2006. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7(1):92.

Mengqiu Wang. 2006. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1).

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM.