AUTOENTITY: AUTOMATED ENTITY DETECTION FROM MASSIVE TEXT CORPORA

BY

WENQI HE

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Jiawei Han

ABSTRACT

Entity detection is one of the fundamental tasks in Natural Language Processing and Information Retrieval. Most existing methods rely on human annotated data and hand-crafted linguistic features, which makes it hard to apply the model to an emerging domain. In this paper, we propose a novel automated entity detection framework, called AutoEntity, that performs automated phrase mining to create entity mention candidates and enforces lexico-syntactic rules to select entity mentions from candidates. Our experiments on real-world datasets in different domains and multiple languages have demonstrated the effectiveness and robustness of the proposed method.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would first like to thank my advisor Professor Jiawei Han of the Department of Computer Science at University of Illinois at Urbana-Champaign. He has always helped me on my research and gave me valuable suggestions on this thesis and other research projects. He also provided me with a research assistant position, so that I could be fully dedicated into my research. I would also appreciate his big support for my Siebel Scholarship. It was my great honor to join his research group. I would also like to thank Jingbo Shang, who gave me a lot of help in my thesis. I sincerely appreciate the help from my colleagues in Data Mining Group of UIUC, especially Xiang Ren, who collaborated with me in my previous studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Entity detection is the task of identying a word or phrase as entity mentions within a text. The extracted entity information can be a great asset for various tasks such as information extraction [1] and knowledge base (KB) population [2].

Traditional supervised machine learning methods [3, 4] for entity detection use fully annotated documents and a variety of linguistic features to train models. To obtain an effective model for a reasonably large domain-specific corpus, the amount of manually annotated data will be significant, which can be costly and time consuming. In addition, such named entity recognition systems [3, 4] are usually designed for general domains (e.g., news), and so require extra and expensive adaptation to a new domain.

Rule-based methods [5, 6] start with a small degree of supervision (e.g., a small set of entities as seeds), create rules from seed entities and use them to incrementally extract new entity mentions and new rules unrestricted by specific domains, which can largely reduce the amount of required labeled data. Rules are typically defined as patterns around the entities, such as lexico-syntactic surface word patterns [7] and dependency tree patterns [8]. Rule-based systems have dominated the commercial world [9], mainly because rules are easy to understand, debug and be incorporated with domain experts. They have also shown superior performance compared to state-of-the-art machine learning methods on some specific domains [10, 6, 11]. However, it still requires human experts to provide initial seeds and will suffer from low recall with sparse context.

In this paper, we study the problem of automated entity detection in a domain-specific corpus: given a domain-specific corpus, we aim to effectively and efficiently detect entity mentions from that corpus without human labeled training data and with minimal linguistic features. We propose a novel automated entity detection framework AutoEntity in this paper, which tries to integrate quality phrase mining together with lexico-syntactic surface word patterns. Quality phrase mining is the task of automatically extracting salient phrases from a given corpus. A recent study of quality phrase mining [12], called AutoPhrase, presents a robust and efficient framework to mine quality phrases from large domain-specific text, with minimal human efforts and reliance on linguistic analyzers. We apply the methodology in AutoPhrase to generate entity mention candidates and create lexico-syntactic surface word patterns automatically from text using a quality measure close to the one introduced in [13]. Then we propose a simple but effective greedy algorithm to enforce learned lexico-syntactic surface word patterns as constraints to refine phrase candidates into entity mentions. To further get rid of additional manual labeling effort, we use external public knowledge bases to generate seed entities for lexico-syntactic pattern learning. As demonstrated in our experiments, AutoEntity not only works effectively in multiple domains like scientific papers, news articles, and discussion forum, but also supports multiple languages, such as English, Spanish, and Chinese.

The main contributions are as follows:

- We study an important problem, automated entity detection, and analyze its major challenges as above.
- We propose a robust lexico-syntactic surface word pattern guided entity detection framework.
- We demonstrate the robustness and accuracy of our method and show improvements over prior methods, with results of experiments conducted on two real-world datasets in different domains (scientific papers, news articles, and discussion forum) and different languages (English, Spanish, and Chinese).

The rest of the paper is organized as follows. Section 2 positions our work relative to existing works. Section 3 defines basic concepts including lexico-sytactic rules and four requirements of quality phrases. The details of our method are covered in Section 4. Extensive experiments and case studies are presented in Section 5. We conclude the study in Section 6.

CHAPTER 2

RELATED WORK

In this chapter, we make an overview of relevant methods and concepts for named entity recognition and quality phrase mining. First in Section 2.1, we introduce studies for named entity recognition and discuss their advantages as well as disadvantages. In Section 2.2, literature about phrase mining is introduced to understand what is phrase mining and state-of-the-art phrase mining algorithms. Then in Section 2.3, we present phrasal segmentation models, which are used to segment a string of words into a sequence of phrases.

2.1 Named Entity Recognition

The task of Named Entity Recognition (NER) is to identify token spans as entity mentions in documents and assign type labels to them. In this section, various named entity recognition methods are discussed in three broad categories of machine learning paradigm. In the first part, we discuss various supervised techniques. Subsequently we move to semi-supervised and unsupervised techniques. In the end we discuss about the method from deep learning to solve NER.

2.1.1 Supervised methods

Traditional supervised methods use fully annotated documents and different linguistic features to train a machine learning model. Hidden Markov Model is the earliest model applied for solving NER problem by Bikel et al. [14] for English. Borthwick [15] and Curran [16] applied the Maximum Entropy Models to the named entity problem. McNamee and Mayfield [17] tackled the NER problem as binary decision problem and used Support Vector Machines as classifiers. McCallum and Li [18] proposed a feature induction method for Conditional Random Fields (CRF) in NER. Later the Stanford NER also adopts a CRF classifier [19]. To obtain an effective model, a large annotated corpus is needed [4] and thus needs heavy human annotation.

2.1.2 Semi-supervised and Unsupervised methods

Semi-supervised learning algorithms typically start with a small set of entities as seed data set and create more labeled entities using large amount of unlabeled corpus. Pattern-based bootstrapping [6, 5] learns patterns from context that identify more entity mentions and new patterns in a bootstrapping cycle but often suffers from low recall and semantic drift.

A major problem of both traditional supervision and semi-supervision is the requirement of annotated data and a robust set of features. Many languages do not have annotated corpus available at their disposal. To deal with lack of annotated text across domains and languages, unsupervised techniques for NER have been proposed. KNOWITALL is an unsupervised system proposed by Etzioni et al. [20] that automatically extracts information from the web in a domain-independent, and scalable manner.

Recently, distantly supervised methods avoid expensive human labeling by leveraging type information of entity mentions which are confidently mapped to entries in knowledge bases. Linked mentions are used to label those unlinkable ones in different ways, including training a label classifier [21, 22], and serving as seeds in graph-based label propagation [23, 24].

2.1.3 Deep learning methods

State-of-the-art named entity recognition systems rely heavily on hand-crafted features and domain-specific knowledge. With the rapid development of deep learning, several research works have been done on applying deep learning methods to the NER task. Recent RNN-based approaches include ones by Lample et al. [25] and Athavale et al. [26]. They both extended a bidirectional LSTM.

2.2 Quality Phrase Mining

Automated extraction of quality phrases (i.e., multiword semantic units) from massive, dynamically growing corpora has become ever more critical due to its value in text analytics of various domains.

As the origin, there have been extensive studies on quality phrase mining in the Natural Language Processing (NLP) community. By leveraging predefined part-of-speech (POS) rules, one can locate noun phrases as term candidates in POS tagged documents. Supervised noun phrase chunking approaches [27, 28, 29] automatically learn rules from annotated documents to identify noun phrase boundaries. To further boost the precision, more sophisticated NLP features (e.g., dependency parser) can be applied [30, 31]. The various kinds of language-dependent linguistic processing and expensive human annotations make it challenging to extend these methods to different domains and languages.

Data-driven approaches have been proposed to take advantage of frequency statistics in document collections. Most of them leverage a variety of statistical measures derived from a corpus to estimate phrase quality. Therefore, they do not require linguistic features, domain-specific language rules or large annotations, and can process massive corpora efficiently. In [32], several indicators from frequency measures have been proposed to extract concepts from large corpora. Deane [33] proposed a nonparametric, rank-based heuristic measure over frequency distribution, for measuring the lexical association for candidate phrasal terms. As a preprocessing step towards topical phrase extraction, ElKishky et al. [34] performed phrase mining based on frequency and proposed a significant measure for bottom-up phrasal segmentation. Jingbo et al. [35] proposed a framework SegPhrase that extracts quality phrases integrated with phrasal segmentation. The segmentation-integrated approach is developed to further rectify the raw frequency scores. A recent work [12] has extended SegPhrase to work automatically without any human effort (e.g., setting domain-sensitive thresholds).

2.3 Phrasal Segmentation

Formally, phrasal segmentation aims to partition a sequence of words into disjoint subsequences each mapping to a semantic unit, i.e., word or phrase. In terms of identifying semantic units, existing work includes query segmentation [36, 37], phrase chunking [38, 39, 40], and Chinese word segmentation [41, 36], following either supervised setting on labeled data, or unsupervised setting on large corpus. Tan and Pang [42] proposed a generative model in unsupervised setting with n-gram frequency from a large corpus and used expectation maximization for computing segment scores. Li et al. [37] exploited query click-through data and proposed a probabilistic model for query segmentation.

CHAPTER 3

PROBLEM DEFINITION

The goal of this paper is to develop an automated entity detection method to extract entity mentions from a large collection of documents without human annotations, and with only limited, shallow linguistic analysis. The input to the automated entity detecting task is a corpus and a knowledge base. The input corpus is a collection of docuemnts in a particular language and a specific domain. The output is a list of detected entity mentions in the corpus. In this section, we briefly introduce basic concepts and components as preliminaries.

First we give definitions of entity mentions and quality phrases. Note that in text corpora, a quality phrase is not necessarily to be an entity mention because there are no syntactic restrictions for phrases (e.g., noun phrases) while an entity mention has a high probability to be a quality phrase because an entity mention by itself is a complete semantic unit and meets the four criteria of quality phrases as introduced below.

Definition 1. *An entity mention is defined as a sequence of words that appear consecutively in text documents which refers to a real-world entity.*

Definition 2. *A phrase is defined as a sequence of words that appear consecutively in text documents, which forms a complete semantic unit in certain contexts of the given documents.*

The phrase quality is defined to be the probability of a word sequence being a complete semantic unit, which meets the following four criteria [35]:

- Popularity: The frequency of a quality phrase should be beyond certain threshold in the given document collection.

- Concordance: The collocation of tokens in quality phrases occurs with significantly higher probability than expected probability assuming independence.

- Informativeness: A phrase is informative if it is indicative of a specific topic or concept.

- Completeness: Long frequent phrases and their subsequences within those phrases may both satisfy the 3 criteria above. A phrase is deemed complete when it can be interpreted as a complete semantic unit in some given document context. Note that a phrase and a subphrase contained within it, may both be deemed complete, depending on the context in which they appear. For example, "relational database system", "relational database" and "database system" can all be valid in certain context.

We follow the approaches in [7, 6] to define lexico-sytactic surface word patterns. To increase the coverage of rules, we also include POS patterns of seed entities as valid lexico-sytactic surface word patterns.

Definition 3. *A lexico-sytactic surface word pattern is defined as a template of context words (optional) around a seed entity and POS tags of the seed entity.*

In the initial stage of lexico-sytactic rule learning, we perform entity linking [43] to automatically generate seed entities.

Definition 4. *The Entity Linking task is the task of automatically linking each named entity mention appearing in a source text document to its unique entry in a target knowledge base.*

CHAPTER 4

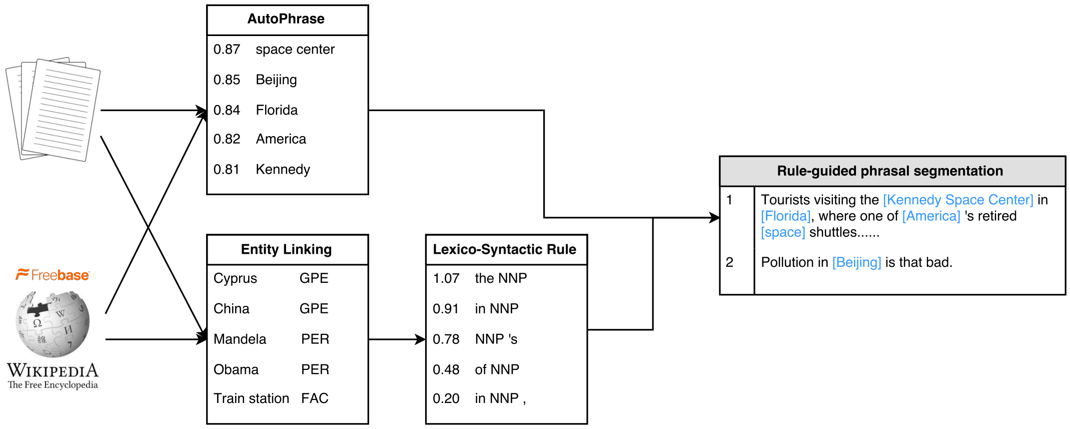
METHODOLOGY

We first present the full procedure of our proposed entity detection framework **AutoEntity**. Then we introduce each of them in following subsections.

1. Perform automated phrase mining on a corpus to extract entity mention candidates. (Section 4.1)

2. Collect seed entity mentions as labels by linking extracted candidate mentions to the knowledge base and use seed entity mentions to generate lexico-syntactic rules. (Section 4.2)

3. Apply a lexico-syntactic rule guided phrasal segmentation to extract entity mentions. (Section 4.3)



AutoPhrase	
0.87	space center
0.85	Beijing
0.84	Florida
0.82	America
0.81	Kennedy

Entity Linking	
Cyprus	GPE
China	GPE
Mandela	PER
Obama	PER
Train station	FAC

Lexico-Syntactic Rule	
1.07	the NNP
0.91	in NNP
0.78	NNP 's
0.48	of NNP
0.20	in NNP ,

Rule-guided phrasal segmentation	
1	Tourists visiting the [Kennedy Space Center] in [Florida], where one of [America] 's retired [space] shuttles......
2	Pollution in [Beijing] is that bad.

Figure 4.1: The overview of AutoEntity.

An illustration for this workflow is shown in Figure 4.1. An complexity analysis for this framework is given at Section 4.4 to show that its computation time grows linearly as the corpus size increases.

4.1 Automatic Phrase Mining

To ensure the extraction of informative and coherent entity mentions, we apply an automated phrase mining method called AutoPhrase [12] to generate entity mention candidates. Almost all the state-of-the-art methods require domain and linguistic experts at certain levels but AutoPhrase requires no manual efforts and is scalable with massive text corpora.

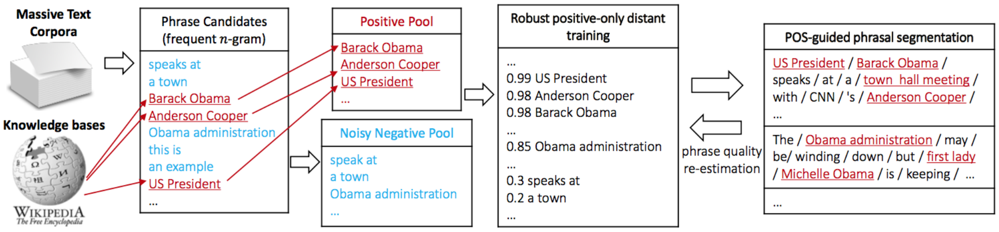

Massive Text Corpora
Knowledge bases
WIKIPEDIA The Free Encyclopedia
Phrase Candidates (frequent n-gram)
speaks at
a town
Barack Obama
Anderson Cooper
Obama administration
this is
an example
US President
...
Positive Pool
Barack Obama
Anderson Cooper
US President
...
Noisy Negative Pool
speak at
a town
Obama administration
...
Robust positive-only distant training
...
0.99 US President
0.98 Anderson Cooper
0.98 Barack Obama
...
0.85 Obama administration
...
0.3 speaks at
0.2 a town
...
phrase quality re-estimation
POS-guided phrasal segmentation
US President / Barack Obama / speaks / at / a / town hall meeting / with / CNN / 's / Anderson Cooper / ...
The / Obama administration / may / be/ winding / down / but / first lady / Michelle Obama / is / keeping / ...
...

Figure 4.2: The overview of AutoPhrase.

The AutoPhrase framework is shown in Figure 4.2. To automatically mine these quality phrases, the first phase of AutoPhrase (see leftmost box in Figure 4.2) establishes the set of phrase candidates that contains all n-grams over a minimum support threshold (e.g., 30) in the corpus. Here, this threshold refers to raw frequency of the n-grams calculated by string matching. In practice, one can also set a phrase length threshold (e.g., 6) to restrict the number of words in any phrase. Given a phrase candidate $w_1 w_2 \ldots w_n$, its phrase quality is:

$$Q(w_1 w_2 \ldots w_n) = p(\lceil w_1 w_2 \ldots w_n \rfloor | w_1 w_2 \ldots w_n) \in [0, 1]$$

where $w_1 w_2 \ldots w_n$ refers to the event that these words forms a quality phrase $\lceil w_1 w_2 \ldots w_n \rfloor$. $Q(\cdot)$ is defined as the phrase quality estimator. We initialize $Q(\cdot)$ with statistical features (e.g., inverse document frequency, point-wise mutual information, and point-wise KL divergence) computed from data. Note that no POS tag information is used for computing the phrase quality estimator $Q(\cdot)$. For unigrams, we simply set their phrase quality as 1.

Example 1. *A good quality estimator will return* $Q(\textit{this man}) \approx 0$ *and* $Q(\textit{international space center}) \approx 1$,

The second phase of AutoPhrase is to estimate the phrase quality score for each phrase candidate by positive-only distant training. The positive-only distant training is a new technology introduced in AutoPhrase, which utilizes public knowledge bases to provide a positive phrase pool and a negative phrase pool. More details could be found in the paper.

Then, to address the completeness criterion, the third phase is a POS-guided phrasal segmentation, which finds the best segmentation for each sentence by incorporating POS tag information. It adopts a generative process to generate a quality segment given a sequence of words and the corresponding POS tag sequence.

During the last phase, phrase quality re-estimation, related statistical features will be re-computed based on the rectified frequency of phrases, which means the number of times that a phrase becomes a complete semantic unit in the identified segmentation.

In AutoEntity, we will use the phrase quality estimator $Q(\cdot)$ returned by AutoPhrase to score the phrase quality of a entity mention candidate.

4.2 Lexico-Syntactic Rule Learning

We learn lexico-syntactic surface word patterns from unlabeled text starting with seed dictionaries of entities. We refer to a sequence of words that represents an seed entity as positive examples. We will present the approach below for learning lexico-syntactic surface word patterns.

Seeding. Entity linking is applied here to map possible entity mentions in the training corpus to a public knowledge base. After entity linking, we could collect seed entities as a seed dictionary. Entity mentions that could be mapped to an entity entry in the knowledge base are considered as seeds. We then utilize the seed dictionary to tag words in the corpus.

Creating rules. Candidate rules are created using contexts of words in a window of two to four words before and after a tagged sequence of words. The target term has a part-of-speech (POS) restriction, which is the POS tags of the tagged sequence of words. That is, given a token t, a literal rule r is generated using a context window of width $w = 3$ around the token and its POS tags:

$$r = [w_{-3}w_{-2}w_{-1}POS(t)w_{+1}w_{+2}w_{+3}]$$

where $w_{\pm i}$ are the context words of t. Two literal rules are generated for each name occurrence of seed entities, one for the left context, and one for the right. The literal rule r is then generalized by replacing some of the words in the context window by wildcards. The generalized rules form the set of candidate rules. Note that each rule matches on only one side of an instance, the left or the right. To further increase the coverage of rule matching, we also include the raw POS tag sequence of the labeled token t as a candidate rule.

Example 2. *Suppose we set the context window width to 3, given the seed entity "America" and sentence: "Tourists visiting the Kennedy Space Center in Florida, where one of* ***America*** *'s retired space shuttles...". We could extract the following candidate rules {NNP, of NNP, one of NNP, where one of NNP, NNP 's, NNP 's retired, NNP 's retired space}.*

Evaluate rules. For every candidate rule r, we match r against the training corpus. Wherever the context of r matches, r predicts the occurrence of possible entity token span. The token span t can be:

- positive example: appears as a entity in the seed dictionary;
- negative example: not included in the seed dictionary.

For each candidate rule r, we compile two lists of tokens matched by r: the positive, and negative examples, or $P(r)$ and $N(r)$. We then compute the rule's confidence:

$$conf(r) = \frac{|P(r)|}{|P(r)| + |N(r)|}$$

Rules with confidence below a threshold are discarded. The remaining rules are ranked by:

$$S(r) = conf(r) \times log|P(r)| \tag{4.1}$$

Thus, to get a positive score, a rule must have at least two distinct token spans as positive examples, and more positive than negative examples. The n top-scoring rules are selected as accepted lexico-syntactic rules and plus their scores are used for further steps in AutoEntity. Domain and language experts could be involved in this step to further improve the quality of mined rules.

4.3 Rule-Guided Phrasal Segmentation

The proposed Rule-Guided Phrasal Segmentation addresses the challenging of locating all phrase mentions mined in Section 4.1 in the corpus and select only entity mention phrases guided by lexico-syntactic rules learned in Section 4.2.

Compared to the POS-Guided Phrasal Segmentation in AutoPhrase [12], the rule-guided phrasal segmentation addresses the completeness requirement by incorporating surrounding context and syntactic constraints, instead of only utilizing POS tags. In addition, lexico-syntactic rules provide shallow, language-specific, and domain-specific knowledge, which may help boost entity detection accuracy, especially at syntactic constituent boundaries for that language. Our method adopts a significance score to guide the filtering of non-entity mention phrases. We partition sentences in the corpus into non-overlapping segments and select segments which meet a significance threshold as entity mentions.

In order to combine the quality of a phrase and the quality of the rules matched by the phrase, we define the significance score in the following way. Given a token span t, we compute its phrase quality $Q(t)$ and fetch its matched rule set R_t from the accepted lexico-syntactic rule set. We define the significance score of the phrase as follows:

$$Score(t) = (Q(t) + s(\sum_{r \in R_t} S(r)))/2 \tag{4.2}$$

More complicated formulas could be applied here to further improve the performance.

Then we develop an efficient greedy algorithm for the rule-guided phrasal segmentation as shown in Algorithm 1. The input of the algorithm is a sequence of words $w_1 w_2 \dots w_n$, accepted lexico-sytactic rules R with a scoring function $S(\cdot)$, a phrase quality estimator $Q(\cdot)$, the maximum phrase length l, and a threshold θ for determining whether the phrase should be a valid entity mention. The output of the algorithm is a list of detected entity mentions.

In the greedy algorithm, we try to partition the sequence of words from left to right and merge words into phrases once a certain criteria is matched. At each iteration, the algorithm looks at all possible phrases starting at current word w_i and select the phrase with the maximum significance score as defined in Equation 4.2. If the maximum significance score is larger than the threshold θ_e, we will tag this phrase as an entity mention. The pseudocode of the greedy algorithm is shown in Algorithm 1.

Algorithm 1 PhrasalSegmentationGreedy($w_1 w_2 \dots w_n, R, S(\cdot), Q(\cdot), \theta_e$)

```
Require: $w_1 w_2 \dots w_n$, the word sequence; $R$, $S(\cdot)$, the set of accepted lexico-sytactic rules and its scoring function; $Q(\cdot)$, the phrase quality estimator; $l$, maximum phrase length; $\theta$, a threshold for determining whether the phrase should be a valid entity mention.
Ensure: $entitylist$, a list of detected entity mentions.
function PhrasalSegmentationGreedy($w_1 w_2 \dots w_n, R, S(\cdot), Q(\cdot), theta_e$)
    $entitylist \leftarrow []$
    $i \leftarrow 1$
    while $i \leq n$ do
        $b_i \leftarrow i$
        for $j \in \{i, \dots, \min(i + l, n)\}$ do
            $t \leftarrow w_i \dots w_j$
            $max\_score \leftarrow -1$
            $R_t \leftarrow$ fetch accepted rules based on $R, t$
            $score \leftarrow$ calculate significance score using $t, R_t, S(\cdot), Q(\cdot)$ according to Equation 4.2
            if $score > max\_score$ then
                $max\_score \leftarrow score$
                $b_i \leftarrow j$
            end if
        end for
        $i \leftarrow b_i + 1$
        if $max\_score > \theta$ then
            $entitylist.add(w_i \dots w_{b_i})$
        end if
    end while
    return $entitylist$
end function
```

4.4 Complexity Analysis

The time complexity of each component in our framework, i.e., automated phrase mining, lexico-syntactic rule learning, and phrasal segmentation, is $O(|\Omega|)$ with the assumption that the maximum number of words in a phrase is a small constant (e.g., $l \leq 6$), where $|\Omega|$ is the total number of words in the corpus. Therefore, AutoEntity is linear to the corpus size and thus being very efficient and scalable.

CHAPTER 5

EXPERIMENTS

In this section, we will apply the proposed method to extract entity mentions from two text corpora across three different domains (new papers, discussion forums, and biomedical paper abstracts) and in three languages (English, Spanish, and Chinese). We compare the proposed method with many other methods to demonstrate its competitive performance. We first explore the adaptiveness of the proposed method in different languages. Then we try to prove the proposed method could be applied to other domains. In the end, we present case studies.

5.1 Datasets

To validate that the proposed method, AutoEntity, can support multiple languages and can effectively work in different domains, we use two real-world datasets in different domains and languages, as shown in Table 5.1:

- **KBP**[44]: It consists of 90,003 documents of news articles (NW) or discussion forum (DF) in three languages (English, Spanish, and Chinese). 500 test documents are manually annotated with five target types (person, location, organization, geo-political entity, facility). We will refer to datasets of each language as **KBP-EN** for English, **KBP-ES** for Spanish and **KBP-CN** for Chinese.

- **PubMed**[45]: It consists of 1M sampled PubMed paper abstracts as training data and 1100 annotated PubMed Abstracts from the CYP corpus of PennBioIE as test data.

Datasets	**KBP-EN**	**KBP-ES**	**KBP-CN**	**PubMed**
Language	English	Spanish	Chinese	English
Domain	NW and DF	NW and DF	NW and DF	Paper abstract
Training documents	29,834	29,832	29,834	1,000,000
Training file size	64M	67M	53M	711M
Test documents	168	168	167	1100
Gold entity mentions	9231	6964	8845	34446

Table 5.1: Statistics of the datasets.

5.2 Compared Methods

We compare AutoEntity with four lines of methods as follows.

- **CRF** [19]: a CRF classifier, the state-of-the-art entity recognition approach used in Stanford CoreNLP toolkit. We used the entity linking results (Section 5.3) as training data to train a binary CRF model, which predicts whether a token belongs to an entity. We used the CRFClassifier code in the latest Stanford CoreNLP toolkit [1].

- **Pattern** [6]: a state-of-the-art pattern-based bootstrapping method which uses a set of initial seed entities, and then extracts new patterns and new entity mentions iteratively. We used the linked entities from entity linking results as initial seed set and extracted new entities in the whole corpus including both training and testing documents. I also used the code in the latest Stanford CoreNLP toolkit [2].

- **ClusType** [24]: a relation phrase based entity recognition method, which runs data-driven phrase mining to generate entity mention candidates and relation phrase candidates, and performs type propagation with relation phrases and multi-view relation phrase clustering simultaneously. We used the code published in GitHub [3].

- **AutoPhrase** [12]: an automated phrase mining method using robust positive-only distant training and POS-guided phrasal segmentation. We used the training corpus to extract quality phrases and learned a segmentation model. Then we applied the segmentation model to test

[1] `https://stanfordnlp.github.io/CoreNLP/`

[2] `https://nlp.stanford.edu/software/patternslearning.html`

[3] `https://github.com/shanzhenren/ClusType`

documents with a tuned threshold to extract entity mentions. In this case, all quality phrases were assumed to be entity mentions. We used the code published in GitHub [4].

5.3 Experiment Settings

Implementation. The preprocessing includes tokenization, Chinese word segmentation and POS tagging from Stanford NLP. To avoid human annotation, we expoited external public knowledge base to automatically label training corpora. We utilized Diffbot API [5], to identify entity mentions from text and map them to the primary entities at DBpedia. We put linked entity mentions back to training documents and created annotated training datasets. We used these automatically annotated training corpora for CRF, Pattern, ClusType and our own method AutoEntity as supervision. In our own method, we used the AutoPhrase code published in Github [6] to generate entity mention candidates. The rest of the implementation will be released and maintained in GitHub after we finish the integration with AutoPhrase.
Parameter Settings. We follow the same parameter setting for AutoPhrase [12]. We set the minimum support threshold as 30. The maximum number of words in a phrase is set as 6. We set the entity threshold for AutoEntity as 1. These are three parameters required by AutoEntity. For fair comparison, the minimum support threshold and the maximum number of words in a phrase are set the same for ClusType. Other parameters required by compared methods were set according to the open-source tools or the original papers.
Evaluation Metrics. Recognizing entity mentions can be seen as a tagging task. For a list of predicted entity mentions, *precision* is defined as the number of true entity mentions divided by the number of predicted entity mentions; *recall* is defined as the number of true entity mentions divided by the total number of golden entity mentions. Here evaluation treats an annotation as a set of distinct tuples, and calculates *precision* and *recall* between

[4] https://github.com/shangjingbo1226/AutoPhrase
[5] https://www.diffbot.com/
[6] https://github.com/shangjingbo1226/AutoPhrase

gold (G) and system (S) annotations:

$$\begin{aligned} P &= \frac{|G \cap S|}{|S|} \\ R &= \frac{|G \cap S|}{|G|} \end{aligned}$$

We also reported the F1 score for entity detection, which is defined as the balanced harmonic mean of P and R:

$$F1 = \frac{2PR}{P + R}$$

5.4 Entity Detection in Different Languages

Table 5.2, 5.3 and 5.4 summarize the comparison results on the KBP dataset for three languages. Overall, AutoEntity outperforms other methods on F1 scores, and achieves competitive precision and recall scores compared to best baselines. Machine learning based approaches, including CRF and Pattern, tend to achieve high precision but low recall, while data-driven approaches, ClusType and AutoPhrase suffer from low precision/high recall.

AutoPhrase performs the best, in terms of recall and F1 scores on both KBP-EN and KBP-ES datasets. For example, on the KBP-ES dataset, the F1 score of AutoEntity is about 7.29% higher than the second best method (AutoPhrase) in relative value. Meanwhile, there is a visible recall gap between AutoEntity and baselines on KBP-EN and KBP-ES datasets. CRF and Pattern have very high precisions on both datasets but suffer from low recalls. For example, on the KBP-ES dataset, AutoEntity achieves a recall 23.9% higher than Pattern in absolute value. This is because context and syntactic features are sparse and AutoEntity makes use of quality phrases from AutoPhrase, which has a high coverage of golden entity mentions. On both datasets, AutoPhrase is very competitive. But the precision of AutoEntity is 8.7% higher than AutoPhrase in absolute value because AutoEntity takes advantages of lexico-syntactic rules.

Significant advantages can be observed on the KBP-CN dataset. In particular, AutoEntity obtains a 14.02% improvement in F1 score compared to the best baseline Pattern, while it maintains close precision to the best baseline

Pattern and close recall to the best baseline AutoPhrase. Data-driven approaches like AutoPhrase and ClusType suffer from low precision on Chinese because Chinese phrases vary and the mined phrases are not constrained to be noun phrases, which is a necessary condition of entity mentions. This is the main reason why AutoPhrase and ClusType achieve high recall but very low precision. AutoEntity utilizes lexico-sytactic rules to overcome this problem. By averaging the quality scores of both lexico-syntactic rules and phrases, AutoEntity obtains superior performance.

Method	Precision	Recall	F1
CRF [19]	**0.794**	0.383	0.517
Pattern [6]	0.605	0.457	0.520
ClusType [24]	0.421	0.500	0.457
AutoPhrase[12]	0.508	0.554	0.530
AutoEntity	0.595	**0.571**	**0.583**

Table 5.2: Performance comparison of entity detection on **KBP-EN** dataset.

Method	Precision	Recall	F1
CRF [19]	**0.802**	0.322	0.459
Pattern [6]	0.729	0.337	0.461
ClusType [24]	0.448	0.417	0.432
AutoPhrase[12]	0.428	0.547	0.480
AutoEntity	0.466	**0.576**	**0.515**

Table 5.3: Performance comparison of entity detection on **KBP-ES** dataset.

Method	Precision	Recall	F1
CRF [19]	0.813	0.311	0.449
Pattern [6]	**0.831**	0.452	0.585
ClusType [24]	0.276	0.581	0.375
AutoPhrase[12]	0.166	**0.598**	0.260
AutoEntity	0.781	0.583	**0.667**

Table 5.4: Performance comparison of entity detection on **KBP-CN** dataset.

5.5 Entity Detection across Domain

We also tested the performance of AutoEntity on a dataset of a different domain. Table 5.5 summarizes the precision, recall and F1 scores of AutoEntity and baselines on PubMed dataset. AutoEntity still achieves the best F1 score compared to other methods while Pattern achieves the highest precision and AutoPhrase achieves the highest recall. AutoEntity tends to achieve a balance between precision and recall compared to Pattern and AutoPhrase, which has a bias towards either precision or recall.

Method	Precision	Recall	F1
CRF [19]	0.356	0.395	0.374
Pattern [6]	**0.386**	0.343	0.363
ClusType [24]	0.279	0.507	0.36
AutoPhrase[12]	0.144	**0.692**	0.238
AutoEntity	0.309	0.536	**0.392**

Table 5.5: Performance comparison of entity detection on **PubMed** dataset.

5.6 Case Study

We present a few case studies about the output of AutoEntity. Table 5.6 shows two example sentences from KBP-EN and PubMed dataset. AutoEntity extracts all entity mentions in the golden mention set. The false positive examples are still reasonable because both KBP and PubMed datasets have a target entity type set which does not cover these false positive examples. However, they could be considered as entity mentions given a more general definition of entities.

Extracted quality phrases are shown in Table 5.7 on KBP-EN and KBP-ES datasets. The top ranked phrases are mostly named entities, which is consistent to our assumption for AutoPhrase that a large proportion of quality phrases are entity mentions. In fact, we have more than 39K and 54K phrases with a phrase quality higher than 0.5 on the KBP-EN and KBP-ES datasets respectively. This ensures the high recall for AutoEntity. Table 5.8 shows the extracted lexico-syntactic rules on KBP-EN and KBP-CN datasets. Most of the extracted rules provide reasonable constraints for locating an entity

Datasets	KBP-EN
Text	**Chinese** legend, Chang e is a **moon** goddess, accompanied by a Jade Rabbit ...
AutoEntity	Chinese, Chang, moon, Jade Rabbit

Datasets	PubMed
Text	Analysis of inhibition in pathways of **NADP.H2** and **NAD.H2** oxidation in liver tissue microsomes.
AutoEntity	inhibition, NADP.H2, NAD.H2, liver, tissue

Table 5.6: Sample output of **AutoEntity** on **KBP-EN** and **PubMed**.

KBP-EN	**KBP-ES**
Rafael Nadal	Broadcasting Corporation
Christine Lagarde	Adolf Hitler
Santa Clara	Morgan Freeman
Walt Disney	Harrison Ford
Serena Williams	Ana Mato
San Lorenzo	Mahatma Gandhi
Mitt Romney	Manny Pacquiao
Saddam Hussein	Florentino Pérez
Santa Claus	Julian Assange
San Marino	Golden State Warriors
Pink Floyd	Antonis Samaras
Silicon Valley	Ink Inc
Silvio Berlusconi	Lionel Messi
Tampa Bay	Pálvaro Uribe
Jimmy Fallon	Christine Lagarde

Table 5.7: Extracted Phrases on **KBP-EN** and **KBP-ES** dataset.

mention. We have more than 23K and 31K rules with a quality score higher than 1 on the KBP-EN and KBP-CN datasets respectively.

5.7 Efficiency Evaluation

To study the time efficiency, we choose the three KBP datasets. Table 5.9 shows the time in seconds for CRF and AutoEntity. AutoEntity achieves about 8 to 14 times speedup compared to CRF, which demonstrates the efficiency of our method.

KBP-EN	**KBP-CN**	Translation
in tomorrow 's NNP NNP NNP	NR 北京	NR Beijing
tomorrow 's NNP NNP NNP	NR 快讯	NR newspaper
republic or its NN	NR 莫斯科	NR Moscow
NN NN mena reported	NR 华盛顿	NR Washington
told NNP	NN 原创 作品	NN original works
NNP pay	NR 东京	NR Tokyo
, official NN NN	NR 难民	NR refugee
communist party of NNP	NR 联邦 储备 委员会	NR Federal Reserve Board
in NNP	NR 巴黎	NR Paris
NNP world news summary	NR 商品 交易所	NR commodities exchange
being sent to NNP	NR NN 政治局	NR NN political bureau
premier NNP NNP	NR 记者	NR journalist
secretary of state NNP NNP	NR 总统 奥巴马	NR President Obama
party of NNP	NR 冬奥会	NR Winter Olympics
cooperation with NNP	俄罗斯 总统 NR	Russian president NR

Table 5.8: Extracted Rules on **KBP-EN** and **KBP-CN** dataset.

Method	**KBP-EN**	**KBP-ES**	**KBP-CN**
CRF	10164.5	12838.1	9765.3
AutoEntity	1237.4	1270.5	694.6

Table 5.9: Time Comparison of **CRF** and **AutoEntity** on **KBP** dataset.

CHAPTER 6

CONCLUSION

In this paper, we present an automated entity detection framework which performs automated phrase mining integrated with lexico-syntactic rule learning. A domain-agnostic phrase mining algorithm, AutoPhrase, is applied for generating entity mention candidates. We create lexico-syntactic surface word patterns automatically around the context of seed entity mentions, which are provided by entity linking. By integrating lexico-syntactic surface word patterns with automated phrase mining, the proposed method is effective in preserving the high recall from automated phrase mining while achieving reasonable precision by posing learned rules on mention candidates. Our experiments show that AutoEntity is domain-independent, and outperforms other entity detection methods, and supports multiple languages (e.g., English, Spanish, and Chinese) effectively. For future work, it is interesting to apply AutoEntity to more languages and perform entity typing on detected entities.

REFERENCES

[1] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* Association for Computational Linguistics, 2012, pp. 523–534.

[2] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2014, pp. 601–610.

[3] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[4] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the 13th Conference on Computational Natural Language Learning.* Association for Computational Linguistics, 2009, pp. 147–155.

[5] S. Shi, H. Zhang, X. Yuan, and J.-R. Wen, "Corpus-based semantic class mining: distributional vs. pattern-based approaches," in *Proceedings of the 23rd International Conference on Computational Linguistics.* Association for Computational Linguistics, 2010, pp. 993–1001.

[6] S. Gupta and C. D. Manning, "Improved pattern learning for bootstrapped entity extraction," in *Proceedings of the 18th Conference on Computational Language Learning.* Association for Computational Linguistics, 2014, pp. 98–108.

[7] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics-Volume 2.* Association for Computational Linguistics, 1992, pp. 539–545.

[8] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Automatic acquisition of domain knowledge for information extraction," in *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 2000, pp. 940–946.

[9] L. Chiticariu, Y. Li, and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, no. October. Association for Computational Linguistics, 2013, pp. 827–832.

[10] R. Nallapati and C. D. Manning, "Legal docket-entry classification: Where machine learning stumbles," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 438–446.

[11] S. Gupta, D. L. MacLean, J. Heer, and C. D. Manning, "Induced lexico-syntactic patterns improve information extraction from online medical forums," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 902–909, 2014.

[12] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *arXiv preprint arXiv:1702.04457*, 2017.

[13] R. Yangarber, W. Lin, and R. Grishman, "Unsupervised learning of generalized names," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[14] D. M. Bikel, R. Schwartz, and R. M. Weischedel, "An algorithm that learns what's in a name," *Machine learning*, vol. 34, no. 1, pp. 211–231, 1999.

[15] A. Borthwick, "A maximum entropy approach to named entity recognition," Ph.D. dissertation, New York University, 1999.

[16] J. R. Curran and S. Clark, "Language independent ner using a maximum entropy tagger," in *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 164–167.

[17] P. McNamee and J. Mayfield, "Entity extraction without language-specific resources," in *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, 2002, pp. 1–4.

[18] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons,” in *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003-Volume 4.* Association for Computational Linguistics, 2003, pp. 188–191.

[19] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2005, pp. 363–370.

[20] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Unsupervised named-entity extraction from the web: An experimental study,” *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[21] X. Ling and D. S. Weld, “Fine-grained entity recognition,” in *Proceedings of the 26th Conference on Artificial Intelligence.* Association for the Advancement of Artificial Intelligence, 2012.

[22] N. Nakashole, T. Tylenda, and G. Weikum, “Fine-grained semantic typing of emerging entities,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2013, pp. 1488–1497.

[23] T. Lin, Mausam, and O. Etzioni, “No noun phrase left behind: detecting and typing unlinkable entities,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.* Association for Computational Linguistics, 2012, pp. 893–903.

[24] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han, “Clustype: Effective entity recognition and typing by relation phrase-based clustering,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2015, pp. 995–1004.

[25] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” *arXiv preprint arXiv:1603.01360*, 2016.

[26] V. Athavale, S. Bharadwaj, M. Pamecha, A. Prabhu, and M. Shrivastava, “Towards deep learning in hindi ner: An approach to tackle the labelled data scarcity,” *arXiv preprint arXiv:1610.09756*, 2016.

[27] K.-h. Chen and H.-H. Chen, "Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation," in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 1994, pp. 234–241.

[28] E. Xun, C. Huang, and M. Zhou, "A unified statistical model for the identification of english basenp," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2000, pp. 109–116.

[29] V. Punyakanok and D. Roth, "The use of classifiers in sequential inference," in *Advances in Neural Information Processing Systems 14.* MIT Press, 2001, pp. 995–1001.

[30] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, 2005, pp. 523–530.

[31] T. Koo, X. Carreras Pérez, and M. Collins, "Simple semi-supervised dependency parsing," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2008, pp. 595–603.

[32] A. Parameswaran, H. Garcia-Molina, and A. Rajaraman, "Towards the web of concepts: Extracting concepts from large datasets," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 566–577, 2010.

[33] P. Deane, "A nonparametric method for extraction of candidate phrasal terms," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2005, pp. 605–613.

[34] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *Proceedings of the VLDB Endowment*, vol. 8, no. 3, pp. 305–316, 2014.

[35] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data.* ACM, 2015, pp. 1729–1744.

[36] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing chinese word segmentation for machine translation performance," in *Proceedings of the 3rd workshop on statistical machine translation.* Association for Computational Linguistics, 2008, pp. 224–232.

[37] Y. Li, B.-J. P. Hsu, C. Zhai, and K. Wang, “Unsupervised query segmentation using clickthrough for information retrieval,” in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval.* ACM, 2011, pp. 285–294.

[38] E. F. Tjong Kim Sang and S. Buchholz, “Introduction to the conll-2000 shared task: Chunking,” in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7.* Association for Computational Linguistics, 2000, pp. 127–132.

[39] G. Blackwood, A. De Gispert, and W. Byrne, “Phrasal segmentation models for statistical machine translation,” in *In Coling 2008: Companion volume: Posters and Demonstrations.* Association for Computational Linguistics, 2008, pp. 19–22.

[40] H. Echizen-ya and K. Araki, “Automatic evaluation method for machine translation using noun-phrase chunking,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2010, pp. 108–117.

[41] R. Sproat, W. Gale, C. Shih, and N. Chang, “A stochastic finite-state word-segmentation algorithm for chinese,” *Computational linguistics*, vol. 22, no. 3, pp. 377–404, 1996.

[42] B. Tan and F. Peng, “Unsupervised query segmentation using generative language models and wikipedia,” in *Proceedings of the 17th international conference on World Wide Web.* ACM, 2008, pp. 347–356.

[43] W. Shen, J. Wang, and J. Han, “Entity linking with a knowledge base: Issues, techniques, and solutions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2015.

[44] H. Ji and J. Nothman, “Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end kbp,” in *Proceedings of Text Analysis Conference.* TAC, 2016, pp. 347–356.

[45] S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, L. Ungar, S. Winters, and P. White, “Integrated annotation for biomedical information extraction,” in *Proceedings of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2004, pp. 61–68.