2015

Term Extraction and Disambiguation for Semantic Knowledge Enrichment: A Case Study on Initial Public Offering (IPO)

Jie Tao
Dakota State University

Omar F. El-Gayar
Dakota State University

Amit Deokar
Dakota State University

Yen-Ling Chang
Dakota State University

Recommended Citation

Tao, J., El-Gayar, O. F., Deokar, A. V., & Chang, Y. (2015, January). Term Extraction and Disambiguation for Semantic Knowledge Enrichment: A Case Study on Initial Public Offering (IPO) Prospectus Corpus. In 2015 48th Hawaii International Conference on System Sciences (pp. 3719-3728). IEEE.

Term Extraction and Disambiguation for Semantic Knowledge Enrichment: A Case Study on Initial Public Offering (IPO) Prospectus Corpus

Jie Tao
Dakota State University
jtao16065@pluto.dsu.edu

Omar F. El-Gayar
Dakota State University
Omar.El-Gayar@dsu.edu

Amit V. Deokar
Pennsylvania State University
Amit.Deokar@psu.edu

Yenling Chang
Dakota State University
Yenling.chang@dsu.edu

Abstract

Domain knowledge bases are a basis for advanced knowledge-based systems, manually creating a formal knowledge base for a certain domain is both resource consuming and non-trivial. In this paper, we propose an approach that provides support to extract, select, and disambiguate terms embedded in domain specific documents. The extracted terms are later used to enrich existing ontologies/taxonomies, as well as to bridge domain specific knowledge base with a generic knowledge base such as WordNet. The proposed approach addresses two major issues in the term extraction domain, namely quality and efficiency. Also, the proposed approach adopts a feature-based method that assists in topic extraction and integration with existing ontologies in the given domain. The proposed approach is realized in a research prototype, and then a case study is conducted in order to illustrate the feasibility and the efficiency of the proposed method in the finance domain. A preliminary empirical validation by the domain experts is also conducted to determine the accuracy of the proposed approach. The results from the case study indicate the advantages and potential of the proposed approach.

1. Introduction

Studies related to advanced knowledge systems and their applications (such as business analytics) have taken a substantial shift due to the proliferation of unstructured data (such as free texts). To analyze such unstructured data, text mining approaches are being studied that enhances traditional data analytics methods with linguistic techniques. The quality of these text mining methods largely relies on the availability and quality of the domain knowledge [1]. The specifications of domain knowledge bases, such as ontologies, are widely used as a means for formally representing machine-readable semantic knowledge from a certain domain [2]. Ontologies typically contain domain terms in a hierarchical, explicitly defined fashion. Such terms are linked by relationships, either *taxonomic* or non-taxonomic. Based on domain knowledge (such as *tacit* knowledge from domain experts, or explicit knowledge of domain theories), well defined ontologies (concise and accurate) are able to assist, or enhance the knowledge intensive analytical processes [3].

However, manually constructing and enriching ontologies are resource consuming and labor intensive activities [4]. To overcome such shortcoming(s), automated learning techniques, as a gateway to (semi-) automatically create domain knowledge bases using machine learning techniques, have been well studied. Information Extraction (IE) has been proven to be one of the most important techniques in terms of enriching ontologies. Among the various sources for IE, text mining turns out to be one of the most effective approaches [5]. An important application of text mining in that regard is to extract and disambiguate relevant terms from domain-specific corpus for enriching knowledge bases. Even though several automated enriching approaches have been developed in recent studies [6]–[9], current approaches rely highly on the size and the quality of the text corpora (as the training set) annotated against formal domain knowledge. Moreover, sometimes the extracted terms are vague or confusing, or too narrow to be used as a reasoning/analytical basis. Thus, two research gaps need to be bridged are: *noise handling* and *knowledge richness* in the ontology enrichment process. For the consideration of generalizability, we focus on the taxonomical relations between extracted terms in the proposed approach.

The key contributions of this paper are twofold. First, we propose a novel approach that addresses aforementioned research gaps by synthesizing IE phases such as term extraction, domain specific term selection/filtering, Word Sense Disambiguation (WSD), and ontology integration/enrichment. The approach is

1530-1605/15 $31.00 © 2015 IEEE
DOI 10.1109/HICSS.2015.448

aimed at learning relevant terms for updating a formal domain knowledge structure and emphasizes two major issues in terms of ontology learning, namely *quality* and *efficiency*. Also, the proposed approach adopts a feature-based method that assists in topic extraction and integration with existing ontologies in the given domain. Second, we present an innovative application of the proposed approach in the finance domain, particularly in the context of a corpus consisting of Initial Public Offering (IPO) prospectuses. A research prototype of this application is reported to illustrate the feasibility and the efficiency of the proposed method in understanding the Initial Public Offering (IPO) phenomenon. The case study intends to extend a manually created seed concept list with *explicit* and *relevant* terms extracted from a domain-specific document corpus. A preliminary empirical validation by the domain experts is also conducted to illustrate the accuracy and advancements of the proposed approach. The result from the case study indicates the advantages and potentials of the proposed approach.

The remainder of the paper is organized as follows: Section 2 presents the details of the proposed approach as its main steps. Section 3 demonstrates the feasibility of the proposed approach through implementation of a research prototype. A preliminary case study is reported in Section 4, in order to assess the efficiency and quality of the proposed approach through the research prototype. Section 5 discusses recent related studies on term extraction and WSD, in order to highlight the advantageous features of the proposed approach. Section 6 concludes the paper, as well as discusses the limitations and future work of this study.

2. Methodology

The purpose of the proposed approach is to enrich domain ontologies from domain-specific textual resources. As stated above, ontologies can be used as a formal conceptualization for annotating, querying, reasoning, and other analytical purposes. The accuracy and efficiency of ontology-based analyses relies on the quality of the ontology, namely *coverage* and *clarity*. Coverage refers to the completeness of the ontology – i.e. the amount of terms/relations formulated in the ontology, while clarity refers to the explicitness and lucency of the terms in an ontology. The proposed approach is aimed at improving both coverage and clarity of an ontology: for increasing the coverage of the ontology, a mechanism is designed to extract and filter related domain terms from a document corpus in a certain domain; for improving the clarity of the ontology, a WSD method is proposed to reduce the conceptual confusion of the selected terms. Finally, the approach includes a mechanism for aligning the newly discovered terms with existing ontologies. In following subsections, we discuss these aspects in detail.

2.1. Domain Specific Term Extraction and Selection

Researchers have indicated in the literature that noun phrases in texts are roughly term candidates in most cases [6]–[8]. Generally, the term extraction and selection process follows three schools of approaches: 1) corpora based approach; 2) heuristic approaches; and 3) hybrid approaches. Corpora based approaches utilize the Part-Of-Speech (POS) tags and syntactic patterns provided by Natural Language Processing (NLP) tools. An example of the linguistic patterns can be found in [10]: noun phrases match the syntactic pattern of $(\texttt{JJ})^*(\texttt{NN})^+$ are selected as candidates from parsed documents (where JJ refers to an adjective, NN denotes to a noun, * denotes zero or more occurrences (optional), while $^+$ denotes one or more occurrences (required). The drawback of linguistic based approaches is that commonly their results rely largely on the amount of cross-sectional documents in a corpus. Thus, applying these approaches in a less mature domain will possibly result in poor outcomes. On the other hand, heuristic approaches rely on the frequencies/statistical measures of (noun) phrases extracted from the document collection. One of the most important measures is "*term-frequency-inversed-document-frequency*" (*tf***idf*), and its variants (one of them is [11]). Despite the importance and usefulness of aforementioned measures, they are not directly applicable to the current research project; the reason is similar to the discussions in [12], terms with low *tf***idf* scores perform better in domain specific cases. A hybrid based approach is a combination of the former two types of approaches [6].

In this paper, we develop our approach along the lines of a hybrid approach. The approach matches predefined linguistic patterns for term candidate selection and utilizes a domain specific heuristic measure for term filtering. Firstly, we have expanded the aforementioned linguistic pattern for our domain specific term extraction purpose. Note that English language stop words are removed from the term candidates, yet determiners (i.e. *a*, *an*, *the*) are kept for pattern matching purpose. In the current phase of this project, we only capture the noun phrases from the document corpus. The noun phrase patterns (NPPs) in regular expressions (along with examples) are reported in following table (Table 1).

Table 1. Noun Phrase Patterns

NPP	Example
$(\texttt{DT})^*(\texttt{JJ})^*(\texttt{NN})^+$	legal proceedings, profits
$(\texttt{NN})^+(\texttt{IN})^+(\texttt{NN})^+$	strategy of competitors

In Table 1, **DT** denotes determiners, while **IN** denotes prepositions. One point worth noting is that the first NPP has two forms: single-word terms (with only one **NN**) and multiple-word terms. With the term candidates extracted, the proposed approach calculates the filtering measures of term candidates in the specific domain through information along two different lines: heuristic information and domain-specific information. For the heuristic information, we propose a ranking measure as shown in Equation (1).

$$rank(t,d) = \sum_{i=1}^{|t|} \frac{freq(n_i,d)}{\max[freq(t,d)]} \times \log\left(\frac{df(n_i)}{\max[df(t)]} + 1\right) \quad (1)$$

In Equation (1), t is an extracted term candidate, $|t|$ is the number of nouns in t, n_i is the i^{th} noun in t. $freq(n_i,d)$ is the occurrence of n_i in document d. Given TC is the set of all term candidates ($t \in TC$), $\max[freq(t,d)]$ is the highest occurrence in d ($\forall t \in TC$). $df(n_i)$ is the occurrence of n_i in a (domain specific) glossary. If none domain specific glossary exists, then WordNet is used as a domain-independent glossary. $\max[df(t)]$ is the maximum of the occurrence of any t in TC that appears in the glossary. The first part of Equation (1) represents the frequency of the term (FREQ), while the second part of it represents the domain relatedness of the term (DR). With the term candidates sorted based on the ranking measure, users can define the amount of terms needed. For instance, if the user decides to select 100 terms, then the top 100 terms from the sorted list are selected. Alternatively, the user can define a threshold on the ranking measure. For example, if the threshold is 0.6, then any term with $rank(t,d) > 0.6$ is selected.

Moreover, a deep cleansing step is incorporated in the term selection phase in order to enforce the domain relatedness of selected terms. All terms met the aforementioned ranking measure are further filtered through such rules. These rules are encoded based on the analytical purposes based on the terms. For instance, if the terms regarding the geographic locations are not relevant in further analysis, a rule will be encoded and enforced as: `DROP (NP(Token.category = "NN" && Token.kind = "LOC"))`. We have listed the deep cleansing rules used in our case study in Section 4 (Table 2).

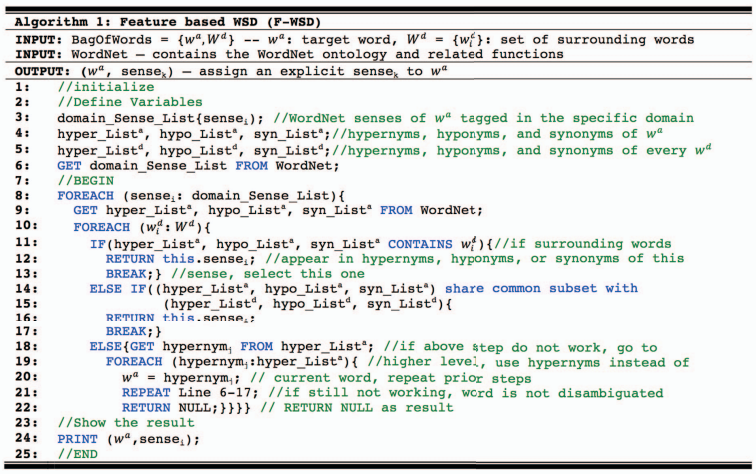
2.2. Word Sense Disambiguation

Word Sense Disambiguation (WSD) is a computational process to identify the explicit meaning of words in a certain context [13]. WSD is an *AI-complete* problem, which means it is among the most difficult problems in the artificial intelligence domain. WSD can enhance the learnt ontologies by reducing the terminological confusion within them [14]. Generally, there are two approaches for WSD: a) learning-based approach; and b) knowledge-based approach. Learning-based approach can be further categorized into *supervised learning based WSD* and *unsupervised learning based WSD*. A key difference between the two is *supervised learning based WSD* relies on tagged documents as a training set for future learning; thus, even though it yields better results, it requires pre-tagged training set – which are usually not available in a less-well-defined domain or on a large sample size. Considering the nature of this project, adopting supervised learning based approaches is not feasible. Knowledge based approach can be further grouped into *dictionary-based* approaches, *corpus-based* approaches and *social media based* approaches. *Dictionary-based* approaches rely on external lexical resources, such as machine-readable dictionaries, thesauri, and ontologies (i.e. WordNet), whereas corpus-based approaches do not use any of them. Instead of using dictionaries, social media based approaches utilize web content (i.e. Wikipedia) as the knowledge base, however, the data quality of such online sources is not guaranteed. In this project, we believe the *dictionary-based* approach is better than the other two.

Before we present the WSD algorithm, we need to discuss the structure of WordNet. We utilize the WordNet taxonomical relations for disambiguating word senses: basically, children classes of current term as hyponyms, parent classes as hypernyms, whereas sibling classes as synonyms. In order to simplify the computation complexity, we limit the scope to direct parent/children classes only.

We present a *feature-based* approach in this paper. Two types of features are adopted in this work, namely *local features* and *syntactic features* [13]. Local features represent a small amount of words around the target word, which their properties such as *POS* tags, word forms, positions; whereas syntactic features represent syntactic information related to the words surrounding the target word. The difference between local features and the syntactic features is that local features are *n-gram* bag-of-words, while syntactic features are features of the words within the same linguistic unit (phrases, sentences, paragraphs, etc.). Words for syntactic features might be outside the *n-gram* bag-of-words. Only words with POS tags of **NN** (nouns), **VB** (verbs), and **JJ** (adjectives) are considered as target words. The feature-based WSD (*F-WSD*) algorithm is presented in Figure 1.

We design the *F-WSD* algorithm based on following design rationale. In a term t, given a surrounding *n-gram bag-of-words* W^d, the target word w^a can be disambiguated if: i) w_i^d ($\exists w_i^d \in W^d$) appears in the hyponyms, hypernyms, or synonyms of w^a; or ii) if hyponyms, hypernyms, or synonyms of w^a and w_i^d



Algorithm 1: Feature based WSD (F-WSD)

INPUT: BagOfWords = $\{w^a, W^d\}$ -- w^a: target word, $W^d = \{w_i^d\}$: set of surrounding words
INPUT: WordNet – contains the WordNet ontology and related functions
OUTPUT: (w^a, $sense_k$) – assign an explicit $sense_k$ to w^a

```
1:  //initialize
2:  //Define Variables
3:  domain_Sense_List{sense_i}; //WordNet senses of w^a tagged in the specific domain
4:  hyper_List^a, hypo_List^a, syn_List^a;//hypernyms, hyponyms, and synonyms of w^a
5:  hyper_List^d, hypo_List^d, syn_List^d;//hypernyms, hyponyms, and synonyms of every w^d
6:  GET domain_Sense_List FROM WordNet;
7:  //BEGIN
8:  FOREACH (sense_i: domain_Sense_List){
9:    GET hyper_List^a, hypo_List^a, syn_List^a FROM WordNet;
10:   FOREACH (w_i^d: W^d){
11:     IF(hyper_List^a, hypo_List^a, syn_List^a CONTAINS w_i^d){//if surrounding words
12:       RETURN this.sense_i; //appear in hypernyms, hyponyms, or synonyms of this
13:       BREAK;} //sense, select this one
14:     ELSE IF((hyper_List^a, hypo_List^a, syn_List^a) share common subset with
15:              (hyper_List^d, hypo_List^d, syn_List^d){
16:       RETURN this.sense_i;
17:       BREAK;}
18:     ELSE{GET hypernym_1 FROM hyper_List^a; //if above step do not work, go to
19:       FOREACH (hypernym_i:hyper_List^a){ //higher level, use hypernyms instead of
20:         w^a = hypernym_i; // current word, repeat prior steps
21:         REPEAT Line 6-17; //if still not working, word is not disambiguated
22:         RETURN NULL;}}}} // RETURN NULL as result
23: //Show the result
24: PRINT (w^a,sense_i);
25: //END
```

Figure 1. The F-WSD Algorithm

shares common subset(s); or iii) if former two conditions are not met, substitute w^a with one of its *direct* hypernyms instead, and repeat previous step. If none of the three conditions is met, the algorithm will return a null value indicating that no disambiguation suggestion can be provided based on given feature values. Essentially, the three conditions listed above can be recognized as classification rules within a logical sequence; thus, decision trees can be used to represent them, which are used to recursively partition the data set. In this context, the data set would be the words in the selected terms requiring disambiguation; the branches are the states in the disambiguation process, the nodes reflect aforementioned conditions, while the leaves are the senses (or *null* value if none sense is selected).

2.3. Ontology Integration

The ontology integration process includes two sub-steps, namely term enrichment and seed concept expansion. We enrich the selected and disambiguated terms with the synonyms/acronyms from the same domain (i.e. "*negative revenues*" and "*losses*"). Further, similar to the term enrichment approach reported in [6], [12], we design a mechanism in the light of enriching multi-word terms. The differences between our method and the methods in [6], [12] are: i) we use a post-selection enrichment, which would reduce the computation complexity of the term extraction and selection phase; and ii) we rely on the ranking of the term based on their DRs from the second part of Equation (1), and then enrich the nouns with rankings higher than a pre-defined threshold – rather than traversing through all the nouns in the selected terms. For instance, given a term ($\mathbf{NN}_1$, $\mathbf{NN}_2$, $\mathbf{VB}_1$, $\mathbf{VB}_2$, $\mathbf{NN}_3$), as well as a pre-defined threshold at 0.7 – if DR($\mathbf{NN}_1$) = 0.8, DR($\mathbf{NN}_2$) = 0.9, and DR($\mathbf{NN}_3$) = 0.6; only $\mathbf{NN}_1$ and $\mathbf{NN}_2$ are chosen for enrichment. The next step is to expand the seed concept list with the selected terms. There are several ways of updating existing ontology with newly discovered terms (i.e. [7], [9]). In this project, we proposed a semantic similarity based approach for such purpose. This approach relies on a similarity matrix, in which each cell represents the similarity between a newly discovered term t_n and an existing term t_e in the seed concept list. Semantic similarity has been widely applied in NLP and Information Retrieval domains, which is termed as a measure of semantic relatedness reflects the semantic relationship (such as "*is-a*" or "*a-kind-of*") based on information theory [15]. A large number of measures with respect to semantic similarity has been published in the literature [15]–[17], which can be categorized as *corpora-based* and *knowledge-based* metrics (a detailed discussion of such categorization can be found in [18]). *Corpora-based* metrics rely on the co-occurrence of a pair of terms within the document corpus, while *knowledge-based* metrics map the terms representing concepts in a formal knowledge structure (such as WordNet or other domain ontologies). The *knowledge-based* measures are more preferable in this work since they rely on knowledge networks rather than (enormous) document corpus or (implicit) external knowledge [19]. A shortcoming of *knowledge-based* approach is that if a term cannot be mapped to the knowledge structure, the measure of semantic similarity is impossible. However, this is not an issue in this project because i) we are updating ontologies – such terms can be treated as new classes in the existing ontology, and ii) the WSD phase reduces, if not eliminates, the possibilities of the "*lack-of-mapping*" issue. Among various *knowledge-based* semantic similarity metrics, we select the *WuP* measure rather than the others – this particular metric measures the normalized depth of concepts and their

Least Common Subsume (LCS) [20]. The rationale behind such design decision is that *WuP* relies on relative depth, and it is normalized, thus it allows working with extremely complex ontology (such as US-GAAP, WordNet, or the Gene Ontology), as well as enables the comparisons across different ontologies. The *WuP* metric is calculated as follows:

$$sim_{WuP}(w_1, w_2) = \frac{2 \times depth(LCS, root)}{depth(w_1, LCS) + depth(w_2, LCS) + 2 \times depth(LCS, root)} \quad (2)$$

In Equation (2), *LCS* refers to the farthest shared parent of a pair of words (ω_1, ω_2) according to the knowledge structure, whereas *root* denotes the root node in it (i.e. *Thing* in WordNet); *depth* is the number of intermediate nodes between two nodes. Figure 2 illustrates an example: $depth(\omega_1, LCS) = 2$, $depth(\omega_2, LCS) = 4$, while $depth(LCS, root) = 1$. Based on Equation (2), $sim_{WuP}(\omega_1, \omega_2) = \frac{2 \times 1}{2+4+2 \times 1} = 0.25$.

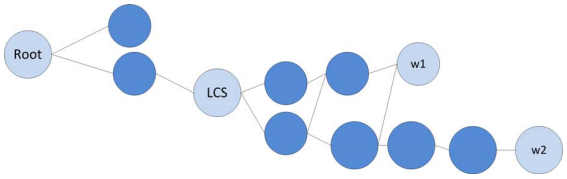


Figure 2. *WuP* Calculation Example

Moreover, the original *WuP* measure reflects the semantic relatedness of two single words. However, in order to align two terms, we need a measure to calculate similarity between multiple-word terms. Thus, we propose the *Normalized Multiple Word Semantic Similarity* (NMWSS) as follows (Equation 3), where $t_n = \{w_i | i = 1 \dots m\}$ and $t_e = \{w_j | j = 1 \dots n\}$ are two multiple-word terms, respectively. If the *NMWSS* between a newly discovered term t_n and an existing term t_e is greater than a pre-defined threshold, then t_n is added to t_e as sub-class/instance; otherwise, a new (sibling) class needs to be created.

$$sim_{multi}(t_n, t_e) = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (sim_{WuP}(w_i, w_j))^2}{m+n}} \quad (3).$$

3. The Design and Demonstration of the Proposed System

The architecture of the proposed system is depicted in Figure 3. In order to deliver a flexible and extensible system, we have adopted *GATE* [21] as the orchestration mechanism for our system. *GATE* is a widely applied NLP toolkit based on Java-like rules (*JAPE, Java Annotations Patterns Engine*), developed for IE and other analytical purposes. *GATE* provides a variety of packaged analytical/processing functionalities (namely Processing Resources, PR), such as *Tokenizer*, *Sentence Splitter*, and *NP Chunker*, for parsing the document corpus. GATE also allows users to encode other functionalities as *JAPE* rules (essentially a pattern-matching *left-hand-side* (LHS) and a Java program as a *right-hand-side* (RHS)), which are executed along with pre-built PRs in a pipeline-like fashion. The corpus itself, along with the domain ontology and WordNet, serves as Language Resources (LRs) in *GATE*. Moreover, other than running standalone, *GATE* can be embedded in other information systems (through provided Application Programming Interfaces, APIs) – so that other elements in the system, such as the User Dashboard, and an independent rule engine for querying/reasoning based on the ontology, can be developed. We implement all three major modules composing the proposed system as *JAPE* rules. Following sub-sections discuss the functionalities of different modules accordingly.

3.1. Term Extraction Module

The *Term Extraction* module undertakes two major functionalities, namely *preprocessing* and *NP extraction*. In the preprocessing of the documents in the corpus, we apply a pre-built plugin in GATE, named *OpenNLP* (Open Natural Language Processing) for parsing the documents. *OpenNLP* is a native plugin incorporated in GATE, originally a library based on *Apache OpenNLP* library [22]. *OpenNLP* also follows a pipeline-like fashion. To make it fitting the purpose of our study, we updated the pre-built package by modifying the code and adding new PRs to it. The major components in the preprocessing sub-module include:

- *OpenNLP Tokenizer*: the *OpenNLP Tokenizer* splits documents into small tokens, such as words, numbers, punctuations, symbols, and spaces.
- *OpenNLP Sentence Splitter*: rather than the default sentence splitter, we select the *OpenNLP Sentence Splitter* in our pipeline and modified the original code to support further segmenting sentences into sub-sentences and/or clauses, which helps us in extracting the syntactic features for term selection purposes.
- *OpenNLP POS Tagger*: the *OpenNLP POS Tagger* assigns POS tags to tokens such as words and symbols with the default lexicon and rule sets. Moreover, the *OpenNLP NER* PR is incorporated in the pipeline, in order to annotate original MUC (Message Understanding Conference) entities, such as *person*, *location*, *organization*, *date*, and so forth. Such annotation is helpful in the later deep cleansing step.

- *Stemmer* and *Morphological Analyzer*: we adopt the two components for lemmatizing the tokens in the document corpus. After this step, all the morphemes (affixes, POS variants, etc.) of the same stem (root words) are annotated with additional features "*stem*" and "*root*" in the token annotations. For instance, a stem feature of "*convert*" is added to both tokens "*converting*" and "*convertible*".
- GATE provides several options in order to implement the *NP Extraction* sub-module, such as *noun phrase chunker* (*NPChunker*), *Tagger Framework*, *LingPipe NER PR*, and the *OpenNLP Chunker*. We select the *OpenNLP Chunker* in implementing the proposed system because: i) as a native PR in the GATE *OpenNLP* plugin, *OpenNLP Chunker* collaborates better with other *OpenNLP* components (such as the ones used in the preprocessing step); 2) as evaluated in a recent study [23], the *OpenNLP Chunker* yields in the highest accuracy and ease-of-use compared to others. The *OpenNLP Chunker* is essentially a JAPE rule – while a rule set is called in the LHS for linguistic pattern matching purposes. We modified the rule set to make sure it fits the linguistic patterns discussed in Section 2.1, while redundant patterns are removed. *OpenNLP Chunker* adds a feature to the tokens in the document, which uses the common BIO values: for instance, a token tagged with a "*B-NP*" value means that it is at the beginning of a noun phrase; while a token tagged with "*I-NP*" means it is inside a noun phrase. This feature is critically useful for identifying the local features as discussed in Section 2.1.

3.2. Term Selection Module

There are two phases in the *Term Selection Module*, namely *related term ranking* and *domain specific deep cleansing*. Several JAPE rules are encoded for implementing this module.

Before calculating the ranking of the term candidates, a linguistic filtering needs to be conducted on them. The first group of JAPE rules is used for such purpose. The first JAPE rule is named "*StopWord-Remove*", which removes the stop words from the extracted term candidates. The English stop word list is obtained from [24]. Then a JAPE rule named "*Filtering*" is added to the pipeline – it has two main functions: filtering tokens with POS tags other than **NN**, **VB**, or **JJ**; and createing the *n-gram bag-of-words* based on the filtered words in the term candidates.

With the term candidates filtered, we can begin to calculate their ranking metrics. The first JAPE rule in this group is named "*Freq-Calculation*", which calculate the term frequency according to the FREQ part of Equation (1). The second JAPE rule "*DR-Calculation*" computes the DR measure, according to the second part of Equation (1). The results from both rules are stored in CSV files. A third JAPE rule "*Ranking*" calculates the final ranking measure, according to Equation (1), and then sort the candidate terms based on the calculated $rank(t, d)$.

The next chunk of JAPE rules conducts '*deep cleansing*' on the words in the sorted term list. The LHS of the JAPE rule "*Deep-Cleansing*" matches the unwanted patterns based on the features from the tokens, while the RHS add a "DROPPED" feature to the corresponding token. A list of exemplar unwanted patterns from the case study can be found in Table 2.

Table 2. Rules Used in Deep Cleansing and Examples

No.	Example //Explanation
1	`(Token.category = "NN" && Token.kind = "LOC") //nouns of geographic locations`
2	`(Token.category = "VB" && Token.chunk = "O") //verbs outside any phrases`
3	`{(Token.category = "CD")}{(Token.category = "NN")} //nouns following a number`

As discussed in Section 2.1, the selection upon the term candidates are completed; and then the selected terms are used as input for the next module.

3.3. Ontology Enrichment Module

Two major phases exist in the *Ontology Enrichment Module*, which respectively are: *word sense disambiguation* and *ontology integration*.

To implement the WSD function in GATE environment, we employed a third-party plugin named *WordNet_Suggester* [25]. In essence, *WordNet_Suggester* is a pre-built JAVA program that provides glossaries, hypernyms, hyponyms, synonyms, and other taxonomic relationships for a specific word, relying on WordNet (which loaded in GATE as a LR). *WordNet_Suggester* provides us a gateway to retrieve the taxonomic information from WordNet, and it allows configuration through initialization parameters (such as *attemptFullMatch*: set to true if intend to match multiple words). However, we have to code a custom JAPE rule in order to realize the WSD method proposed in Section 2.2 (*FeatureWSD*). This rule uses output annotation set from the *WordNet_Suggester* of both the selected terms (outcomes of the *Term Selection Module*) and the *n-gram bag-of-words* surrounding the target word as inputs (LHS patterns), and implements the *F-WSD* algorithm on the RHS. It adds a feature "*WN_sense*" to the target

word (token): if a sense is determined, then the sense is added as the value of "*WN_sense*"; otherwise, a *null* value is added.

The second phase in the *Ontology Enrichment Module* is ontology integration, according to Section 2.3. For calculating the semantic similarity proposed above, we adopt *ws4j* (WordNet Similarity for Java) package [26] by calling its Application Programming Interface (API) in the JAPE rule "*simCal*", which is used to calculate the *NMWSS* (as presented in Equation (3)) between words in two terms (discovered-existing pairs). Another JAPE rule, "*OntoSuggester*", is developed to suggest the expansion of the seed concept list based on the calculation results from "*simCal*" and a user-defined threshold (as a runtime parameter). It is by design that the "*OntoSuggester*" rule does not directly update the concept list; instead it provides suggestions to the users/knowledge workers – in other words, it assists the concept expansion process, rather than replacing human judgments. Then variations of "*simCal*" and "*OntoSuggester*" are executed in the pipeline, between the pair of terms from the expanded concept list and the target ontology (the ontology requiring alignment to).

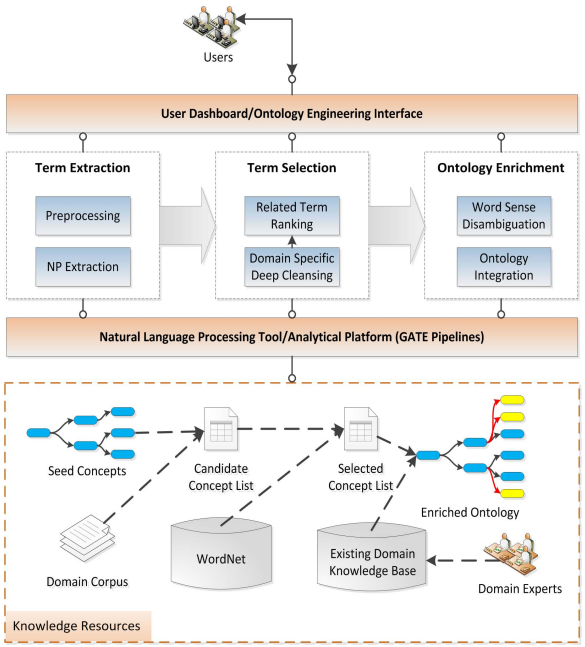


Figure 3. Architecture of the Proposed System

In other words, with the term extracted and disambiguated (described in Section 3.1 and 3.2), we leverage the semantic similarity between the selected terms for the purpose of 'fitting' them in the seed concept list. Thus, knowledge workers can use such enriched, non-ambiguous ontology for querying and other purposes (i.e. reasoning).

4. A Preliminary Case Study

The proposed system is instantiated in a research prototype '*IPO-Extractor*' and evaluated in a case study in the finance domain. The *IPO-Extractor* prototype is deployed on a machine equipped with an Intel Xeon 2.47 GHz CPU, 8GB RAM, and Windows 7 Enterprise 64-bit operating system.

The details regarding the case study are elaborated in following subsections, with some preliminary results and discussions.

4.1. Design of the Case Study

The case study aims to learn the knowledge structure regarding the IPO process through the textual contents in the IPO prospectus. Two domain experts created a seed concept list named the *IPO-Ontology*, which contains the key concepts with respect to the *Risk Factors* section in the prospectus. IPO prospectus is recognized as the most credible source when analyzing the phenomena within an organization's IPO process; whereas the *Risk Factors* section is deemed as one of the most information-rich sections within the prospectus [27]. It is well accepted that the textual information in the *Risk Factors* section has a significant effect on the IPO pricing volatility; yet no formal knowledge structure exists in the domain to support the IE-based analysis on it [27], [28]. We plan to apply the *IPO-Extractor* on document corpus containing the *Risk Factor* sections of the IPO prospectus, for the purpose of enriching the *IPO-Ontology* for further analyses.

IPO-Ontology was developed with 6 first-level classes, and 47 second-level classes in a hierarchical manner. The root concept is "*risk_factors*", and the first level classes include: *growth* (concepts related to the growth and business operations of an organization), *management skills* (the management views and strategies of a company), *competitiveness* (the ability to compete with the competitors), *customers* (the relation with current and potential customers), *lawsuit* (the capability to issue and react to a lawsuit, or potential lawsuit), and *stock prices* (the pricing strategy of the stock), which are the key factors disclosed in the *Risk Factor* sections that affecting the IPO pricing. These factors are generalized from an intensive literature review in the finance domain. A snippet of *IPO-Ontology* (in OWL format) is shown in Figure 4.

In order to obtain the document corpus for the case study, we developed a web crawler to retrieve 424B documents from the EDGAR database of the Security Exchange Committee (SEC). 424B documents are the final prospectus in the IPO process.

Several filtering rules created by the domain experts were applied in the web crawler, such as the company should not be in the finance industry (i.e. banking or insurance companies), the IPO should be within the period between 2003-1-1 to 2013-12-31, and common stock only. Any prospectus without a valid *Risk Factors* section is expunged. More than 2,000 424B documents were retrieved. A random sampling is conducted and a total of 150 documents were selected for this case study.

```
<owl:Class rdf:ID="competitiveness">
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >protections</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >protection</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >competitors</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >competitor</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >compete</altMatch>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="factors_risk"/>
  </rdfs:subClassOf>
  <prefMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >competitiveness</prefMatch>
</owl:Class>
<owl:Class rdf:ID="management_skill">
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >recruiting</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >assimilates</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >attracts</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >personel</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >executives</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >executive</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >personels</altMatch>
  <altMatch rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

Figure 4. A Snippet of *IPO-Ontology*

Specifically for the purpose of the case study, a parsing/pruning step is conducted before applying *IPO-Extractor* on the document corpus. The parsing step aims at removing all non-textual contents in documents (including tables, figures, table of contents, file head, etc.) and annotating the Risk Factors sections from the 424B documents. The average length (number of word tokens) of the selected documents is 84,581; whereas the average length of the Risk Factors sections is 3,874 (4.58% of the average document length).

4.2. Results and Discussions

In this case study, we have selected the first 50 term candidates in the sorted list (using Equation (1)) and set the threshold for NMWSS to 0.5.

To evaluate the competitiveness of our method against other term extraction methods, we have selected several GATE-based term extraction methods, including: ANNIE (A Nearly-New Information Extraction System) + NPChunker, ANNIE + KEA (Key Phrase Extractor). The same document corpus and the threshold of top 50 terms are used for both other methods. We use the duration of the extraction process (Duration, in minutes) and the RAM usage at peak during the extraction process (RAMUSE, in Gigabytes (GB)) as indicators of the *efficiency* of different methods. The results are shown in following table (Table 3). From Table 3, it is clear that Duration and RAMUSE of IPO-Extractor is lower than the other two methods. The duration of IPO-Extractor is 59.9% and 72.8% of the other two methods, while the RAMUSE is 77.5% and 84.6%. These two metrics suggest that the proposed IPO-Extractor is more efficient than current GATE-based term extraction methods.

Table 3. Efficiency Comparison Between Different Methods

Method	Duration	RAMUSE
IPO-Extractor	118 min.	5.5 GB
ANNIE+NPChunker	197 min.	7.1 GB
ANNIE+KEA	162 min.	6.5 GB

For the purpose of testing the *quality*, two domain experts were asked to manually extract terms from 100 out of the 150 selected documents. A total of 57 terms were extracted from the document corpus. Out of the 57 manually extracted terms, 43 appear in the results extracted by *IPO-Extractor*. To compute the values of the evaluation metrics, we define the selected terms as *positives* and the dropped term candidates as *negatives*. Since our data is highly skewed – negatives are much more than positives, a precision/recall/F-measure test is employed. Thus, if we use the manually extracted terms as ground truth, *IPO-Extractor* achieves a precision of 76%, a recall of 66.7%, and an F-score of 71%. The contingency table of the results is shown in Table 4.

Table 4. The Contingency Table of the Results from the Case Study

		Condition	
		Positive	Negative
Test	Positive	38	12
	Negative	19	291

Discussion of the results. As shown in Table 3, in terms of efficiency and resource intensiveness, *IPO-Extractor* is better than other two term extraction methods embedded in GATE. With the fine-tuning of the components, *IPO-Extractor* is more temporally and computationally efficient. The enriched ontology evidently enhanced the information extraction process in terms of coverage: the number of sentences extracted from the 'risk factors' section has increased 150% - 300% across the selected sample. Also, domain experts have reported that the enriched ontology is capable of assisting them in determining whether an extracted sentence is relevant or not. With respect to the accuracy of *IPO-Extractor*, the precision/recall is comparable to other term extraction methods [7], [23], [29]. Given the fact that we applied an unsupervised method in the term selection phase, and used WordNet, as a generic knowledge structure – rather than a domain specific knowledge structure; the result is fairly encouraging. However, it also points out the possible directions for future study.

5. Related Work

Several previous studies have proposed term extraction methods in various domains [6]–[10], [12], [30], [31]. Contrary to our approach, only two methods used domain specific ranking mechanisms for selecting terms. Moreover, none of the method employed a domain specific cleansing step to further filter the extracted terms. Even though some of these methods achieve better accuracy, it is partially because some of them utilized a strictly tested domain ontology [6] and/or supervised methods with a training data set [31]. Our methods can be applied in domains where no explicit, formal knowledge structure exists (such as the IPO domain illustrated in the case study); or cost-sensitive domains where obtaining a training data set is not feasible. Alternatively, our approach can be adopted as a pilot step in an iterative term extraction process – the ontology enriched by our method can be used as the underlying knowledge structure for further term extraction purposes or other ontology-based analyses, such as information extraction, document clustering, or reasoning.

The approach proposed in this paper also sheds light on WSD. A large body of work has been done in developing/improving WSD methods (a detailed review can be found in [5]). Comparing to them, our approach is novel from two standpoints: i) applying feature-based WSD while using dictionaries is not common in prior methods; ii) our approach suggests the possibility to align domain specific knowledge base(s) (i.e. *IPO-Ontology*) with domain independent knowledge base(s) (i.e. WordNet). From the latter standpoint, our approach potentially enables domain-specific, ontology-based reasoning using axioms and semantic relations inherited from domain independent resources. On the other side of the coin, since WordNet cannot fully reflect all semantic relations from real-world scenarios, the marriage with domain-specific knowledge resources would further enrich WordNet by adding new properties/relations.

6. Conclusion and Future Research

In this paper, we design an approach for enriching ontologies through term extraction, word sense disambiguation, and enrichment phases. The proposed approach is then implemented in a research prototype and evaluated by a case study in the finance domain. The preliminary results indicate that the proposed method is comparable to the state-of-art term extraction methods. Our approach is novel in the sense of the *quality* (knowledge richness and explicitness) and *efficiency* (computational complexity and resource dependency). However, there are several limitations to the current study: i) the relatively small sample size in our case study; ii) the lower precision comparing against other term extraction and ontology enrichment methods; and iii) the intuitive linguistic and domain specific patterns. These limitations point out the future directions of this study.

Other than the abovementioned points, we also plan to take the following steps in the future to further enhance this study:

- Methodology-wise: firstly, we plan to explore other NLP solutions in order to boost the performance of our approach; secondly, domain specific taxonomies will be synergized in our approach for WSD purposes; thirdly, it would be interesting to verify our approach in a supervised and iterative fashion, which can improve the performance of our approach; last but not least, the approach presented in this paper focuses on the hyponymy/synonymy relations between extracted terms – since we use WordNet as the knowledge reference. It will be interesting to *further investigate* other semantic relations formulated in domain knowledge.
- Application-wise: this study is a section of a larger project [32], which aims at studying the IPO pricing strategies based on the IPO prospectus. The approach proposed in this paper prepares the basis for further analysis. In the future, we plan to: i) apply the approach or its improved variants to textual contents in other important sections in the IPO prospectus, and then construct the ontology for the overall IPO field; ii) use the enriched ontology as the basis to extract knowledge from the IPO prospectus, and then use such knowledge for constructing predictive models for understanding the IPO pricing phenomenon.

References

[1] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007, p. 422.

[2] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Creation Diffusion Utilization*, no. April, 1993.

[3] A. Maedche and S. Staab, "Ontology learning for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 72–79, 2001.

[4] L. Zhou, "Ontology learning: state of the art and open issues," *Information Technology and Management*, vol. 8, no. 3, pp. 241–252, Mar. 2007.

[5] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the

future," *ACM Computing Surveys*, vol. 44, no. 4, pp. 1–36, Aug. 2012.

[6] Y.-B. Kang, P. Delir Haghighi, and F. Burstein, "CFinder: An intelligent key concept finder from text for ontology development," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4494–4504, Jul. 2014.

[7] M. Gaeta, F. Orciuoli, S. Paolozzi, and S. Salerno, "Ontology Extraction for Knowledge Reuse: The e-Learning Perspective," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 4, pp. 798–809, Jul. 2011.

[8] A. Ittoo and G. Bouma, "Term extraction from sparse, ungrammatical domain-specific documents," *Expert Systems with Applications*, vol. 40, no. 7, pp. 2530–2540, Jun. 2013.

[9] T. C. Dorji, E. Atlam, S. Yata, M. Fuketa, K. Morita, and J. Aoe, "Extraction, selection and ranking of Field Association (FA) Terms from domain-specific corpora for building a comprehensive FA terms dictionary," *Knowledge and Information Systems*, vol. 27, no. 1, pp. 141–161, Apr. 2010.

[10] Q. Li and Y.-F. B. Wu, "Identifying important concepts from medical documents," *Journal of biomedical informatics*, vol. 39, no. 6, pp. 668–79, Dec. 2006.

[11] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[12] X. Jiang and A. Tan, "Mining Ontological Knowledge from Domain-Specific Text Documents," *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 665–668, 2005.

[13] R. Navigli, "Word sense disambiguation: A Survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, Feb. 2009.

[14] H. Wimmer and L. Zhou, "Word Sense Disambiguation for Ontology Learning," in *Proceedings of the Nineteenth Americas Conference on Information Systems*, 2013, pp. 1–10.

[15] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, vol. 1.

[16] D. Lin, "An information-theoretic definition of similarity," in *Proceeding ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 296 – 304.

[17] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," in *WordNet: An electronic lexical database*, MIT press, 1998, pp. 265–283.

[18] D. Sánchez, M. Batet, A. Valls, and K. Gibert, "Ontology-driven web-based semantic similarity," *Journal of Intelligent Information Systems*, vol. 35, no. 3, pp. 383–413, Oct. 2009.

[19] J. Ge and Y. Qiu, "Concept Similarity Matching Based on Semantic Distance," in *2008 Fourth International Conference on Semantics, Knowledge and Grid*, 2008, no. C, pp. 380–383.

[20] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94)*, 1994, pp. 133–138.

[21] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: an architecture for development of robust HLT applications," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, no. July, pp. 168–175.

[22] Apache-OpenNLP-Development-Community, "Apache OpenNLP Developer Documentation," 2014. [Online]. Available: https://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html.

[23] N. Kang, E. M. van Mulligen, and J. a Kors, "Comparing and combining chunkers of biomedical text," *Journal of biomedical informatics*, vol. 44, no. 2, pp. 354–60, Apr. 2011.

[24] Snowball-Tartarus, "English stop word list," 2013. [Online]. Available: http://snowball.tartarus.org/algorithms/english/stop.txt.

[25] P. Gooch, "GATE plugin for adding WordNet features to annotations," 2013. [Online]. Available: https://github.com/philgooch/WordNet_Suggester.

[26] H. Shima, "WS4J," 2013. [Online]. Available: https://code.google.com/p/ws4j/.

[27] K. W. Hanley and G. Hoberg, "The Information Content of IPO Prospectuses," *Review of Financial Studies*, vol. 23, no. 7, pp. 2821–2864, 2010.

[28] T. Loughran and B. McDonald, "IPO First-Day Returns, Offer Price Revisions, Volatility, and Form S-1 Language," *Journal of Financial Economics*, vol. 109, no. 2, pp. 307–326, Aug. 2013.

[29] J. M. Ruiz-Martínez, R. Valencia-García, J. T. Fernández-Breis, F. García-Sánchez, and R. Martínez-Béjar, "Ontology learning from biomedical natural language documents using UMLS," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12365–12378, Sep. 2011.

[30] C. Zhang, Z. Niu, P. Jiang, and H. Fu, "Domain-specific term extraction from free texts," *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, no. Fskd, pp. 1290–1293, May 2012.

[31] H. Nassif, R. Woods, E. Burnside, M. Ayvaci, J. Shavlik, and D. Page, "Information Extraction for Clinical Data Mining: A Mammography Case Study," in *2009 IEEE International Conference on Data Mining Workshops*, 2009, pp. 37–42.

[32] J. Tao, A. V. Deokar, and O. F. El-Gayar, "An ontology-based information extraction (OBIE) framework for analyzing initial public offering (IPO) prospectus," in *Proceedings of the 47th Annual Hawaii International Conference on System Sciences (HICSS-47 '14)*, 2014.