Finding Additional Semantic Entity information for Search Engines

Jun Hou
Queensland University of Technology
Brisbane
jun.hou@student.qut.edu.au

Richi Nayak
Queensland University of Technology
Brisbane
r.nayak@qut.edu.au

Jinglan Zhang
Queensland University of Technology
Brisbane
jinglan.zhang@qut.edu.au

ABSTRACT

Entity-oriented search has become an essential component of modern search engines. It focuses on retrieving a list of entities or information about the specific entities instead of documents. In this paper, we study the problem of finding entity related information, referred to as *attribute-value pairs*, that play a significant role in searching target entities. We propose a novel decomposition framework combining reduced relations and the discriminative model, Conditional Random Field (CRF), for automatically finding entity-related attribute-value pairs from free text documents. This decomposition framework allows us to locate potential text fragments and identify the hidden semantics, in the form of attribute-value pairs for user queries. Empirical analysis shows that the decomposition framework outperforms pattern-based approaches due to its capability of effective integration of syntactic and semantic features.

Categories and Subject Descriptors

I.2.7 [**Computing Methodologies**]: Natural Language Processing – *Language parsing and understanding; Text analysis*

General Terms

Algorithms, Design, Experimentation.

Keywords

Entity Retrieval, Decomposition Framework, Conditional Random Field (CRF)

1. INTRODUCTION

Due to the rapidly increasing size and wide spread of the Web, it has become an immense knowledge repository which contains rich information of entities and their relations. In parallel, as search engine technologies evolve, Entity Retrieval [1, 21] and Question Answering [26] have become crucial components of modern web information retrieval systems. Entity Retrieval and Question Answering seek to find information of individual entities that meet the expected constraints imposed by users in the form of queries, rather than the documents. The constraints help filter out irrelevant answer candidates and determine the answer selection criteria.

Understanding search constraints at entity level in Web queries has been a focused theme of many research methods from the area of Query Intent Classification [3, 14, 22], Query Modifier identification [18, 19] and Identifying Semantic Structure of Queries [15]. These methods are largely driven by the requirement of entity level searchable information. On the other hand, ontologies (with knowledge on entity level) have been used to boost the performance of entity-oriented search. Several lightweight ontologies that encode hierarchical entity related information have been proposed including Yago [24], Freebase[1], amd DBpeida[2]. Generally, these ontologies contain Class-Instance and Class-Attribute hierarchy at entity level and non-taxonomic entity relations such as “hasWonPrize”. Wikipedia category is used in [25, 28], Yago has been used in [7] and Freebase is used in [6]. However, it is very difficult to find a comprehensive ontology that covers all entity related information for a general domain. For example, Yago and DBpedia target a limited number of non-taxonomic entity relations because of using handcrafted rules [4]. On the other hand, free text in documents holds rich context and linguistic information and contains entity level information implicitly.

In this paper, we focus on the problem of automatically finding semantic information for entities by integrating linguistic information and external domain knowledge. Our goal is to identify all potential entity level information, such as ontology-like Class-Attribute and any non-taxonomic relations for a target entity in free text. Web search engines can then provide more precise results based on the fine-grained semantic information about entities instead of just returning documents based on keywords matching.

Unlike structured data sources, the entity related information in free text is usually formulated as sequences of words without much explicit semantic information. In our work, entity related information is modeled as *attribute-value pairs*, (<*attribute*>, <*value*>). Traditionally, a set of entities consists of an entity class exhibiting a set of properties. These general entity properties inherited from the entity class can be referred to as “*noun phrase attributes”*. However, unlike the pairs of *Class-Attribute* are explicitly represented in domain ontology, attribute-value pairs in free text often exist in an implicit form. For example, the fragment “Australian state of Victoria”, does not contain any segment that corresponds to the attribute name “country” for its value “Australian”. A significant sub-task presented in this paper is identifying the explicit and implicit attributes and their values. Moreover, the attribute-value pair can be described in the form of

ADCS '12, December 5 – 6, 2012, Dunedin, New Zealand.
Copyright 2012 ACM 978-1-4503-1411-4/12/2012 …$15.00.

[1] http://www.freebase.com/

[2] http://dbpedia.org/About

non-taxonomic relation, which we refer to as “*relation attribute*”. For example, the relation “FoundDate” with its value “1785” can be interpreted from the text “Melbourne is founded in 1785”. This type of relation plays a significant role for answering factual questions. For example, identification of a relation between recording company and Kingston Trio's songs, would be vital to answer the query such as “What recording companies now sell the Kingston Trio's songs?”.

In this paper, we propose a decomposition framework by reducing the triple that encodes the relations between attribute-value pairs and entities. The triple $r = \ < entity,\ attribute,\ value >$ is reduced to $r' \ = \ \ < entity,\ class >$ and $r'' \ = < entity|class,$ $value|attribute - value >$. (The “$entity|class$” in r''denotes that either $entity$ or $class$ can be the left argument. Similarly $value|attribute - value$ denotes that either value or attribute-value can be the right argument showing that it is possible to have implicit or explicit attribute for an entity’s value.). The property-denoting attribute-value pairs can be inferred by finding the reduced relations r' and r'', and then by identifying the semantic roles in r' and r'', i.e., $entity, class, attribute\ and\ value$.

In the decomposition framework, the reduced relation r' is first detected from the context text. Once the relation between entity and its class is identified, the task is extended to identify r'' for finding attribute-value pairs. We then propose to apply Conditional Random Fields (CRF) models to assign semantic role for elements in r' and r'', i.e., $entity, class, attribute$ $and\ value$. Both noun phrase attribute and relation attribute will be found in r' and r''.

More specifically, contributions of our work are as follows: (1) Modeling entity level information as attribute-value pairs; (2) Proposing a novel decomposition framework for automatically finding attribute-value pairs; and (3) Presenting methods that identify semantics of attribute-value pairs with the related entity.

Section 2 discusses related work. Section 3 introduces the proposed decomposition framework. Evaluation is presented in Section 4. Section 5 concludes this paper and discusses future work.

2. RELATED WORK

General relations containing class-attribute and the associated entity properties are valuable for building concept representations. The framework proposed in this paper integrates such upper level knowledge as features for finding all possible attribute-value pairs of an entity. The majority of existing entity property studies uses a semi-supervised learning approach to extract class-attribute pairs for an entity [16, 17]. These methods aim to extract general class-attribute information, i.e. finding attributes like, “director” or “cast” for an entity class, “movie”. Researchers have identified class instances (entities) from unstructured text with seed entities and use them to extract attributes from query logs using query templates. In order to provide high coverage and quality class-attribute, lightweight ontologies, such as YAGO [24] and DBpedia have been developed to integrate entity level information from different sources. These ontologies consist of entities grouped into different entity classes and each entity is attached to related attributes and relations. However, these works mainly focus on noun phrase attribute and target a limited number of pre-defined relations.

Another relevant research area to our work of studying semantics of attribute-value pairs in free text is the semantic studies of adjective-noun phrases, i.e., assigning attributes to property-denoting adjectives. For example, in “a blue car”, the hidden attribute “color” should be assigned to the value “blue”. In particular, authors in [11] developed a representation composition framework and utilized structured vector space models (SVSM) to map adjective-noun phrases to attribute semantics. They [12] further proposed an approach using Topic Models of LDA to discover the inherent semantics between the attribute and the adjective. Authors in [10] leverage the Expectation-Maximization (EM) algorithm to learn an attribute-value classifier for a similar task. A key difference with our work from these works is that we extend the attribute-value pair identified from the noun phrase attribute to the relation attribute and focus on connecting attribute-value pairs with the corresponding entity. For instance, we identify the attribute “color” for the value “blue” as well as which entity is related to “a blue car”.

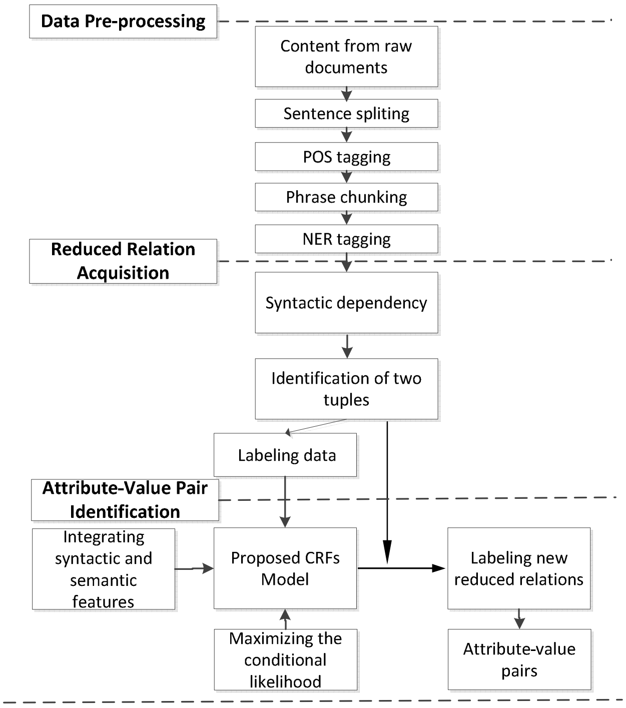
Data Pre-processing
Content from raw documents
Sentence spliting
POS tagging
Phrase chunking
NER tagging
Reduced Relation Acquisition
Syntactic dependency
Identification of two tuples
Labeling data
Attribute-Value Pair Identification
Integrating syntactic and semantic features
Proposed CRFs Model
Labeling new reduced relations
Maximizing the conditional likelihood
Attribute-value pairs

Figure 1. Overview of our proposed decomposition framework.

Another research area that is related to this paper is Open Information Extraction (IE), as finding attribute-value pairs for an entity can be viewed as a specific task discovering reduced relations between attribute-value pairs and entity without pre-defined rules. The task of Open IE was introduced by [8] with a state-of-the-art Web IE system, TEXTRUNNER. The system learns unknown relations based on self-supervised framework using a small set of domain-independent features from the training set. This framework is further extended to utilize different types of CRF such as supervised, self-supervised and stacked for extracting relations [5]. [27] proposes a novel Open IE system based on syntactic dependency representation using the structured sources from Wikipedia Infobox. Second generation Open IE systems, such as Reverb [20] and R2A2 [9] are proposed to further improve extraction performance. The differences between these works and ours are that the open IE systems only deal with explicitly mentioned relations, whereas, we focus on finding all possible reduced relations as well as finding the attribute-value semantics hidden in reduced relations.

3. A DECOMPOSITION FRAMEWORK FOR ENTITY RELATED ATTRIBUTE-VALUE PAIRS

As illustrated in Figure 1, it includes three major steps: (1) pre-processing data; (2) acquiring reduced relations; and (3) training the CRF model and applying it to identify new attribute-value pairs. The CRF model is trained with the reduced relations labeled with semantic roles, such as $entity, class, attribute\ and\ \ value$.

3.1 Data Pre-processing

The first step in the proposed decomposition framework is to traverse over the text corpus for processing each sentence to detect reduced relations. We resort to linguistic processing techniques such as sentence boundary detection, part-of-speech (POS) tagging, phrase chunking and Named Entity Recognition.

Text in each document is first split into sentences using a sentence boundary detection tool[3]. A part-of-speech (POS) Tagger[4] is then used to annotate each sentence with POS tags. After that, we use a phrase chunking tool[5] to group word tokens into phrases. To detect as many potential entities as possible, a Name Entity Recognition (NER) tool[6] is first used and then the heuristic rules discussed in Section 3.2.1 are applied. Once we have completed processing the text corpus, we identify reduced relations as explained in Section 3.2.

3.2 Reduced Relation Acquisition

Ideally, the entity related information can be identified by searching for patterns in the text data [2, 11, 23]. However, linguistic patterns may easily become overfit and have difficulty to find quality information due to a large amount of noise present in the data. In this paper, we utilize the decomposed representation to address the quality issue. The entity related attribute-value pair is modeled as a triple of entity, attribute and value. The triple r is then broken down into two tuples r' and r''as shown in Figure 2.

$$r = \ < entity, attribute,\ value >$$

$$r \ = < r', r'' >$$

$$r' \ = < entity, class >$$
$$r'' \ = < entity|class, value|attribute - value >$$

Figure 2. Reduced relations for triple r.

The "$value|attribute - value$" in r''indicates that it is possible that an entity can have implicit or explicit attribute for its value. These two tuples are modeled as $\{< arg1, rel, arg2 >\}$, where $args$ are any possible pair of left and right arguments of r' or r'' , i.e, entity and class or entity and value, and rel represents the textual fragment indicating semantic relation between two arguments. The reduced relation r' is first detected from the context text. Once the relation between entity and its class is identified, the task is extended to identify the reduced relation r'' for finding attribute-value pairs. Next, we discuss how to find reduced relation r' and r''.

3.2.1 Syntactic dependency

Syntactic dependency representation is designed to provide a description of the grammatical relations. It explains the relations between pairs of words in a sentence. For example, the Stanford parser dependencies of "Quebec City is the capital of the Canadian province of Quebec." are represented as in Figure 3a.

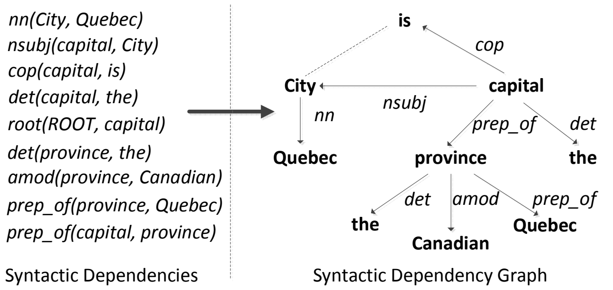

Figure 3. Syntactic dependencies and corresponding syntactic dependency graph.

Each dependency represents a relation between a pair of word tokens, e.g., "nn(City, Quebec)". We form these dependencies into a directed graph G (Figure 3b), $G \ = < V, E >$, where V is a set of nodes containing all word tokens in the sentence, e.g., "City" or "capital", and E are edges denoting the relation between any pair of word tokens, e.g., "nsubj". We then use the shortest connecting path that includes subject, verb and object of a sentence to find relation:

$$City\ \overrightarrow{nsubj}\ capital\ \overleftarrow{cop}\ is$$

We call this as BasicRelation and it is useful to capture information of a basic relation. However, it loses semantic information on phrase level. For example, the BasicRelation does not indicate the integrity of entity "Quebec City". In order to capture meaningful relations, we expand the BasicRelation with phrase level information by adding modifier dependencies of the word tokens in BasicRelation, such as adverbial and adjectival modifiers as well as dependencies that modify verb, like "neg" and "aux". We utilize the expanded BasicRelations to derive our reduced relations r' and r''.

3.2.2 Finding Reduced Relation

Tuple $\mathbf{r'}$. We first examine the related entity with expanded BasicRelation to identify potential reduced relation r'. In $\{< arg1, rel, arg2 >\}$, if $arg1$ or $arg2$ contains the related entity, it is a potential reduced relation r' . We then check if the relation, rel indicates an IsA relation. The relation rel is checked against with IsA relation pattern, IsA($arg1$, $arg2$), which can be summarized as:

(A1) [arg1] copula [arg2]

where *copula* represents any form of copula from the context where target entity appears. For example, the extended BasicRelation, "Quebec City$_{\mathrm{arg1}}$ is$_{\mathrm{copula}}$ capital$_{\mathrm{arg2}}$" carries an IsA relation. If the rel is not an IsA relation, the extended BasicRelation is then put into finding tuple r''.

Tuple $\boldsymbol{r''}$. For $\{< arg1, rel, arg2 >\}$, if $arg1$ or $arg2$ contains related entity or extended class, it is considered as a reduced relation r''. For tuple r'', the rel is not limited (except IsA

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] http://nlp.stanford.edu/software/tagger.shtml

[5] http://nlp.stanford.edu/software/corenlp.shtml

[6] http://nlp.stanford.edu/ner/index.shtml

relation) and contributes to the semantics of attribute-value pair. One example can be, "The city $_{\mathrm{arg1}}$ is founded in $_{\mathrm{rel}}$ 1834_{arg2}". In the example, $arg1$ is an extended class and rel reveals the semantics of attribute-value pair for the extended class.

The text segments, that contain the reduced relation r' or r'', are added to the candidate pool of attribute-value pairs for semantic analysis as explained in next section.

3.3 Attribute-Value Pair Identification

We propose to use a discriminative model, Conditional Random Fields (CRF), for identification of attribute-value semantics in the attribute-value pair's candidate pool. We cast the problem of identifying attribute-value pairs as a joint segmentation or classification problem. Our goal is to semantically tag attribute and value for a related entity in reduced relations r' and r''. The reduced relations are labeled with semantic roles such as $entity, class, attribute\ and\ \ value$.

3.3.1 Model

Conditional Random Fields (CRF). The CRF model, a form of undirected graphical model, is a probabilistic framework for labeling sequential data [13]. Its definition is as follows:

Given a graph $G = (V, E)$ where V denotes the nodes and E denotes the edges. Let $Y = (Y_v)_{v \in V}$ and (X, Y) is a conditional random field conditioned on X when Y_v obeys the Markov property with respect to G. X is a set of observed sequence input and Y is the set of random variables over the corresponding sequence. The probability of a set of labels Y for a sequence X under a linear chain CRF with features is:

$$p(y|x) = \frac{1}{Z(x)} exp\,(\sum_{v \in V,i} \lambda_i\, t_i\ (e, Y \mid e, X) + \sum_{v \in V,i} \mu_i\, s_i\ (v, Y \mid v, X)) \tag{1}$$

Here $Z(x)$ is normalization factor, s_i is a state feature function and t_i is a transition feature function, λ_i and μ_i are corresponding weights. The goal of using CRF is to obtain the marginal distribution of the labels Y given an observed sequence X.

Let $X = (x_1, x_2, \ldots, x_n)$ denote an input reduced relation r' or r'' with word length of *n*. $Y = (y_1, y_2, \ldots, y_k)$ represents the semantic labels of k attribute-value pairs and c is the class of the related entity e. Our goal is to obtain the most probable labelling sequence Y of attribute-value pairs for an input X of text segment:

$$\hat{Y} = \underset{y}{argmax}\, p(y|c, e, x) \tag{2}$$

where the related entity e and its class c is identified in r'. Therefore, equation (2) can be written as:

$$\hat{Y} = \underset{y}{argmax}\, p(y|x) \tag{3}$$

Equation (3) is short for notional simplicity and denoting that the label and parameter space are entity- and class-independent.

3.3.2 Label Scheme

In order to train the proposed CRF model, a label scheme is designed to tag reduced relations. We develop five types of label to tag each word, as shown in Table 1.

Table 1. Label sets and their meaning for the proposed CRF model.

Label	Meaning
E	Related entity
C	Entity class
A	Entity attribute e.g., "population"
V	Value for corresponding attribute
O	Others that do not have above semantics

For attribute (A) and value (V), we further use character "Exp" and "Imp" for explicit value and implicit value, respectively. Explicit value means that the value is explicitly stated with its attribute and implicit value means that the value needs to be induced. A text segment may contain multiple words. We apply position labels to each word in the segment. Any text segment contains two positions: the beginning of the segment (B) and the rest of the segment (I). We assign "O" to words that do not contribute to any semantics. With these tags, reduced relations r' or r'' have been tagged. For example,

$\boldsymbol{r'}$. *Quebec(E-B) City(E-I) is(O) the(O) capital(C-B) of(O) the(O) Canadian(V-Imp-B) province(A-B) of(O) Quebec(V-Exp-B).*

$\boldsymbol{r''}$. *The(E-B) city(E-I) is(O) founded(A-B) in(A-I) 1608(V-Exp-B).*

In the example of r', "Canadian" is an implicit value for attribute "country", while explicit value "Quebec" has a corresponding attribute "province" in the text. For the example of r'', the rel, "founded in" serves as relation attribute and the $arg2$ "1608" is the explicit value.

3.3.3 Model Features

In this section, we explore the integration of rich features, including not only transition features but also syntactic features and semantic features in the CRF model to identify attribute-value pairs.

Transition Features. A transition feature (*Trans*) indicates label transition between adjacent states in CRF. For example, in "Quebec(E-B) City(E-I)", the transition feature captures the label changing from t_{j-1}, "E-B" to t_j, "E-I". We only use the first-order transition feature.

Syntactic Features. The reduced relations, which have certain syntactic style, intend to have attribute-value pairs. We use word features, part-of-speech (POS) features and segment features as syntactic features.

A word feature (*W*) is a binary feature that indicates if a specific word co-occurs with a label. We generalize this feature to *n-grams* by applying a sliding window. Each word of the input sequence $w_{1:N}$ is sequentially viewed as the centre of a window with size *n*. In other words, a word feature inspects current position word as well as *n-grams* identity. In this way, the context word features are explored to consider long distance word dependency. Since a word feature follows the linear order principle, the corresponding POS tag of input word is considered as another syntactic feature. The pos feature (*Pos*) indicates whether a label occurs depending on the part-of-speech of the current word. The part-of-speech feature is also extended from the current word to its neighborhood with a size of *n*.

Based on POS tagging, words are organized into k different segments by phrase chunking. These segments can provide a syntactic clue about that which words are in the same segment and which words are not. We refer to this feature as segment feature (*PC*). These segments are used to learn the co-occurrence between labels and syntactic segments. In other words, a segment feature favors words appearing in the same or an adjacent segment. Furthermore, another type of segment feature (*RI*) is created by capturing segments in a reduced relation. The reduced relation has an inherent structure i.e., $\{< arg1, rel, arg2 >\}$, which we refer to as the self-supervised segment feature.

Semantic Features. Semantic features (*Sem*) concern what a word means and how it is related to attribute or value. We create semantic features based on Named Entity Recognition (NER) and semantic lexicons.

NER is implemented as a semantic feature to express what label a named entity class is related to. For example, if "Sydney Harbor" is labelled as value for attribute "located on", the CRF model captures the entity class "Location" as NER semantic feature. When a new named entity with same class "Location" occurs, it would be labeled as value for attribute "located on". The name entity classes used for NER include: Location, Person, Organization and Misc. We also extend the name entity class feature to neighborhood with the length of n.

Similar to the named entity class features, we create semantic lexicons to generalize semantic features. A lexicon is a list of words/phrases with same semantic meaning. For example, the attributes e.g., "country" or "population" can be grouped into an attribute lexicon. Similarly, a list of country names or state names, also form a value lexicon for corresponding attributes. Generally, the lexicon is built from a structured data table or domain ontological knowledge. However, this type of data source generally contains limited semantic information. In order to enrich semantic lexicons, we apply some heuristics:

$$(h_1) \qquad\qquad \underline{E}\ has|had\ \underline{Attr|Value}$$

The $\underline{E}$ represents an entity and $has|had$ implies the attribute and/or value. The heuristics in h_1 is applied on web-scale data and attributes and values discovered by (h_1) are added to our semantic lexicons. If one semantic lexicon presents in the input data, the semantic lexicon feature will be activated and deactivated if not. To better incorporate semantic lexicons to the CRF model, we relax the exact matching to relatedness matching by measuring similarity between semantic lexicon elements and input data. In this paper, we adopt *Levenshtein distance* for the similarity function,

$$Sim(I_j, SL_i) = 1 - Lev_{distance}/|L| \qquad (4)$$

where I_j and SL_i represent the current word or segment of input data and ith element in the semantic lexicon. $|L|$ is the length of the ith element used to normalize *Levenshtein distance*, $Lev_{distance}$. The semantic element with max similarity score from equation (4) is then used as the semantic lexicon feature.

4. EVALUATION

4.1 Datasets

Two document collections are included in the experiment. One is the general purpose dataset available as Web documents to be used by a search engine. In this paper, we use Google and the dataset can be assumed as Web documents indexed by Google. In the experiment, we randomly select 30 seed entities for each entity class, *city* and *movie*. Each seed entity becomes a query (e.g., "Melbourne"). It is submitted to search engine for obtaining Web documents that contains the seed entity. Due to the fact that the higher rank a document appears in a search result, more relevance it has with the search query, the top k documents (k = 100) are collected. Some examples of seed entities are as follows:

$$Seed\ Entity_{city} = \{Melbourne, New\ York, Boston, Beijing, London, \ldots\}$$

$$Seed\ Entity_{movie} = \{Transformers\ 3, Iceage\ 3, The\ hangover\ 2, \ldots\}$$

Another dataset we used is Wikipedia. Similarly, using a seed entity to query Wikipedia documents, Wikipedia documents that contain seed entities are added to our experimental dataset. After the experimental dataset is pre-processed as discussed in Section 3.1, the proposed decomposition framework is applied to extract candidate sentences that contain seed entities. After removing duplicate and non-related sentences, 1000 reduced relations are collected for city domain and 1000 reduced relations for movie domain. All reduced relations are annotated using labels in Table 1 and split into 90/10 for training/testing of the CRF model.

4.2 Evaluation Metrics

Two evaluation metrics are used: Macro-average F1 (F1) and label accuracy (Acc). F1 is computed based on label precision and recall. More specifically, the precision (P) and recall (R), are calculated as the number of labels divided by the number of true positive labels and the number of correct labels divided by the number of true positive labels, respectively. The Macro-average F1-measure is then measured by precision and recall:

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

Secondly, a label of a word is true positive if the label assigned by the trained CRF model matches with its correct label. Label accuracy is measured by the total number of labels divided by the total number of true positive predicted by CRF model.

4.3 Results and Discussion

4.3.1 Syntactic Features

We organize different features into various feature sets to evaluate the performance of every single feature. The first experiment we did is to evaluate the performance of syntactic features. Although the single feature of POS or W does not perform well, the integration of POS and W feature provides a better average F1 score and label accuracy (54.4/66.9) than using any of these features alone as shown in Table 2. It implies that certain word with its syntactic clue is related to attribute-value pair. Both average F1 score and label accuracy in feature set (4) and (5), compared to feature set (3), obtain absolute gain when adding any segment feature (RI or PC). This indicates that attribute-value pairs co-occur with syntactic segments. Moreover, the PC feature offers a small gain than the reduced relation segment feature, RI. This may be caused by the nature of reduced relation that the relation part, rel, is a verb phrase, which generally includes a preposition word e.g., "in, of" as the end of the rel part. This nature may damage the correct form of syntactic segments. Using all syntactic features and transition features (5) achieves the best performance.

4.3.2 Semantic Features

In this section, we evaluate the performance of semantic feature and its integration with syntactic features. Results in (6) (7) (8) as shown in Table 2 prove the consistency of semantic feature boosting the performance of the CRF model for finding attribute-value pairs. This explains the dependency between attribute-value

Table 2. Macro-average F1 (F1) and label accuracy (Acc) using CRF with different features.

	City		Movie		Average	
Features (%)	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
(1) Trans+W	48.1	61.2	50.5	62.0	49.3	61.6
(2) Trans+Pos	52.9	66.3	54.8	67.2	53.8	66.7
(3) Trans+W+Pos	53.7	66.4	55.3	67.4	54.4	66.9
(4) Trans+W+Pos+RI	56.0	68.7	59.7	70.5	57.8	69.6
(5) Trans+W+Pos+PC	61.8	71.1	64.2	73.7	63.0	72.4
(6) Trans+W+Pos+Sem	68.5	75.1	72.1	79.5	70.3	77.3
(7) Trans+W+Pos+PC+Sem	68.4	74.7	71.9	78.8	70.1	76.7
(8) Trans+W+Pos+RI+Sem	69.0	77.0	72.7	80.6	70.8	78.8
(9) Trans+W+Pos+PC+RI+Sem	**69.4**	**77.1**	**73.5**	**80.9**	**71.4**	**79.0**

pairs and knowledge domain. More specifically, it shows that an entity class contains different attributes and different entity classes implicitly select their inherent attributes and corresponding values. By integrating all syntactic and semantic features with transition features, we obtain the highest performance of the proposed CRF model.

4.3.3 Comparison with Baseline

Previous approaches of identifying attributes and their values focused on pattern-based methods [2, 11, 23]. Generally, they apply a set of heuristic rules to fetch demanded pattern instances. Motivated by [12], we implement a pattern-based method as the baseline for comparison. All reduced relations are tagged with their part-of-speech (pos). We then apply a set of patterns on pos-tagged reduced relations to find entity level information. The pattern-based method consists of two steps, finding related entity with its attributes and then capturing the value for attributes. Two groups of patterns are created to extract binary relations, $rel(related\ entity, attributes)$ and $rel(attributes, value)$. We first discover related entity (E) with its attribute (ATTR) using patterns:

(1) E with|without DT? RB? JJ? ATTR
(2) DT ATTR of DT? RB? JJ? E
(3) DT E's RB? JJ? ATTR
(4) E has|had a|an RB? JJ? ATTR

and then identify related attribute (ATTR) and value (V = {JJ | DT? RB? JJ? NN}) with patterns:

(1) ATTR of DT? E is|was V
(2) DT? RB? V ATTR
(3) DT? JJ or V ATTR
(4) DT? E's ATTR is|was V
(5) is|was|are|were V in|of ATTR

Figure 4 and Table 3 show the comparison between CRF model and the baseline on entity class, city. The result shows that our CRF model with only syntactic features outperforms the pattern-based method. The low recall and high precision of the pattern-based method indicates that pre-defined patterns face the overfitting problem. This also proves the effectiveness and robustness of the proposed decomposition framework and CRF model.

5. CONCLUSIONS

In this paper, we propose a novel decomposition framework integrating reduced relations and the discriminative model, CRF, for finding entity-related attribute-value pairs. In the decomposition framework, we first extract reduced relations and then automatically identify the semantics of attribute-value pairs using the model of CRF. Our experiment shows that the proposed CRF model achieves a high performance by integrating syntactic and semantic features with transition features. The proposed method outperforms the current state-of-art method based on patterns detection. In future, we would explore more semantic features, such as semantic lexicons describing the same reduced relation, and minimum supervision to train the CRF model.

Table 3. Comparison between CRF and a pattern-based method.

	P	**R**	**F1**
Baseline (City) (%)	43.0	11.9	18.6
(1) Trans+Pos	55.3	57.9	52.9
(2) Trans+W	63.9	48.5	48.1
(3) Trans+W+Pos	**60.1**	**58.1**	**53.6**

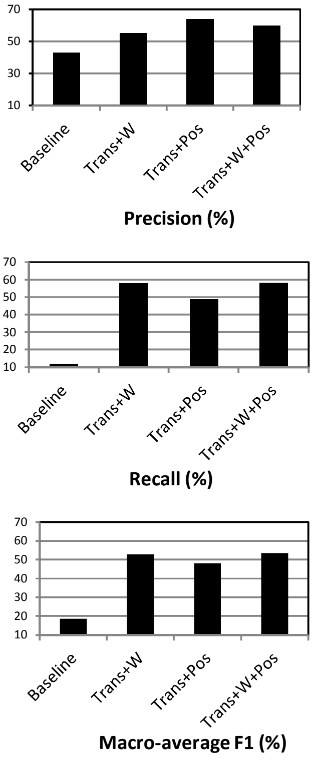

Figure 4. Comparison between CRF models and a pattern-based method.

6. REFERENCES

[1] Adafre, S. F., Rijke, de M., and Sang, E. T. K. 2007. Entity Retrieval. In Proceedings of International Conference of Recent Advances in Natural Language Processing (Borovets, Bulgaria, 2007). RANLP'07. John Benjamins, Amsterdam. Netherland.

[2] Almuhareb, A. 2006. Attributes in Lexical Acquisition. University of Essex, Colchester.

[3] Arguello, J., F. Diaz, F., Callan, J., and Crespo, J. F. 2009. Sources of evidence for vertical selection. In Proceedings of ACM International Conference on Research and development in information retrieval (Boston, MA, USA, 2009). SIGIR'09. ACM, New York, NY, 315–322. DOI= http://doi.acm.org/10.1145/1571941.1571997.

[4] Banko, M. 2009. Open Information Extraction for the Web. University of Washington, Seattle.

[5] Banko, M. and Etzioni, O. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. In Proceedings of Annual Meeting of the Association for Computational Linguistics, (Ohio, USA, 2008). ACL'08. Association for Computational Linguistics, Stroudsburg, PA, 28–36.

[6] Bron, M., He, J., Hofmann, K., Meij, E., Rijke, M. D., Tsagkias, M., and Weerkamp, W. 2011. The University of Amsterdam at TREC 2010: Session, Entity and Relevance Feedback. In Proceedings of Text REtrieval Conference TREC 2010 (Gaithersburg, USA, 2011). TREC'11. NIST Special Publication, Gaithersburg, Maryland.

[7] Demartini, G., C. S. Firan, C. S., lofciu, T., Krestel, R., and Nejdl, W. 2010. Why finding entities in Wikipedia is difficult, sometimes. Inf. Retr, 135, 534–567. DOI= http://doi.acm.org/10.1007/s10791-010-9135-7.

[8] Etzioni, O., M. Banko, M., Soderland, S., and Weld, D. S. 2008. Open information extraction from the web. In Proceedings of International Joint Conference on Artificial Intelligence (Hyderabad, India, 2008). IJCAI'08. AAAI Press, Palo Alto, California, 2670–2676. DOI= http://doi.acm.org/10.1145/1409360.1409378 .

[9] Fader, A., Soderland, S., and Etzioni, O. 2011. Identifying relations for open information extraction. In Proceedings of Conference on Empirical Methods in Natural Language Processing (Edinburgh, United Kingdom, 2011). EMNLP'11. Association for Computational Linguistics, Stroudsburg, PA, 1535-1545.

[10] Ghani, R., K. Probst, K., Liu, Y., Krema, M., and Fano, A. 2006. Text mining for product attribute extraction. ACM SIGKDD Explorations Newsletter, 81, 41–48. DOI= http://doi.acm.org/10.1145/1147234.1147241 .

[11] Hartung, M. and Frank, A. 2010. A structured vector space model for hidden attribute meaning in adjective–noun phrases. In Proceedings of International Conference on Computational Linguistics (Beijing, China, 2010). COLING'10. Association for Computational Linguistics, Stroudsburg, PA, 430–438.

[12] Hartung, M. and Frank, A. 2011. Exploring supervised LDA models for assigning attributes to adjective–noun phrases. In Proceedings of Conference on Empirical Methods in Natural Language Processing (Edinburgh, United Kingdom, 2011). EMNLP'11. Association for Computational Linguistics, Stroudsburg, PA, 540–551.

[13] Lafferty, J. D., A. McCallum, A., and Pereira, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of International Conference on Machine Learning (Williamstown, USA, 2001). ICML'01. Morgan Kaufmann Publishers Inc, San Fransisco, CA, 282–289.

[14] Li, F., X. Zhang, X., Yuan, J.H., and Zhu, X.Y. 2008. Classifying what–type questions by head noun tagging. In Proceedings of International Conference on Computational Linguistics (Manchester, United Kingdom, 2008). COLING'08. Association for Computational Linguistics, Stroudsburg, PA, 481–488.

[15] Li, X. 2010. Understanding the semantic structure of noun phrase queries. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Uppsala, Sweden, 2010). ACL'10. Association for Computational Linguistics, Stroudsburg, PA, 1337–1345.

[16] Pasca, M. 2007. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In Proceedings of International conference on World Wide Web (Banff, Canada, 2007). WWW'07. ACM, New York, NY, 101–110. DOI= http://doi.acm.org/10.1145/1242572.1242587.

[17] Pasca, M. 2008. Turning web text and search queries into factual knowledge: hierarchical class attribute extraction. In Proceedings of National Conference on Artificial intelligence (Chicago, Illinois, 2008). AAAI'08. AAAI Press, Palo Alto, California, 1225–1230.

[18] Pasca, M. and Durme, B. V. 2007. What you seek is what you get: extraction of class attributes from query logs. In Proceedings of International joint conference on Artifical intelligence (Hyderabad, India, 2007). IJCAI'07. Morgan Kaufmann Publishers Inc, San Fransisco, CA, 2832–2837.

[19] Pasca, M. and Durme, B. V. 2008. Weakly–supervised acquisition of open–domain classes and class attributes from web documents and query logs. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Ohio, USA, 2008). ACL'08. Association for Computational Linguistics, Stroudsburg, PA, 19–27.

[20] Reverb. http://reverb.cs.washington.edu

[21] Rode, H. 2008. From document to entity retrieval: improving precision and performance of focused text search. University of Twente, Enschede.

[22] Shen, D., J.–T. Sun, J.T., Yang, Q., and Chen, Z. 2006. Building bridges for web query classification. In Proceedings of ACM International Conference on Research and development in information retrieval (Seattle, USA, 2006). SIGIR'06. ACM, New York, NY, 131–138. DOI= http://doi.acm.org/10.1145/1148170.1148196.

[23] Sowa, John F. 2000. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Distributed Systems Online, 51, 1–3.

[24] Suchanek, F. M., Kasneci, G., and Weikum, G. 2007. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In Proceedings of International World Wide Web Conference (Banff, Canada, 2007). WWW'07. ACM,

New York, NY, 697–706. DOI= http://doi.acm.org/10.1145/1242572.1242667 .

[25] Tsikrika, T., P. Serdyukov, P., Rode, H., Westerveld, T., Aly, D, and Vries, A. P. 2008. Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah. In Proceedings of Focused access to XML documents: 6th international workshop of the initiative for the evaluation of XML (Dagstuhl Castle, Germany, 2008). INEX'08. Springer–Verlag, Heidelberg, Germany, 306–320. DOI= http://dx.doi.org/10.1007/978-3-540-85902-4_27.

[26] Voorhees, E. M. and Harman, D. 2004. Overview of the TREC 2004 Question Answering Track. In Proceedings of Text REtrieval Conference TREC–4 (Gaithersburg, USA, 2004). TREC'04. NIST Special Publication, Gaithersburg, Maryland, 1–11.

[27] Wu, F. and Weld, D. S. 2010. Open information extraction using Wikipedia. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Uppsala, Sweden, 2010). ACL'10. Association for Computational Linguistics, Stroudsburg, PA, 118–127.

[28] Zirn, C., V. Nastase, V., and Strube, M. 2008. Distinguishing between instances and classes in the Wikipedia taxonomy. In Proceedings of European semantic web conference on The semantic web: research and applications (Tenerife, Spain, 2008). ESWC'08. Springer–Verlag, Heidelberg, Germany, 376–387. DOI= http://dx.doi.org/10.1007/978-3-540-68234-9_29.