Lexical Acquisition for Information Extraction from Arabic Text Documents

Mervat Gheith
Dept. of Computer and Information Sciences, ISSR, Cairo University
Cairo, Egypt

Magdy M. Aboul-Ela
Sadat Academy for Management Sciences, Maadi, Cairo, Egypt

Waleed Arafa
Dept. of Computer and Information Sciences, ISSR, Cairo University
Cairo, Egypt

Abstract: - The objective of this work is to design a lexicon suitable for information extraction from Arabic texts, and to acquire this lexicon automatically for specific domain from set of electronic documents. To achieve this goal we have to find a way to represent the document as well as the domain knowledge, extract the document and domain knowledge, then design a lexicon suitable for IE tasks, and fill this lexicon automatically from the acquired information. In this paper, we propose a graph representation for documents and domain knowledge, and IE lexicon which holds domains and its related events and entities with their attributes and extraction patterns to be used for information extraction. We propose also semantic graph representation for documents and extraction patterns for each event and entity within a domain, unsupervised learning algorithm to build those graphs, with a new technique to extract information using those semantic graphs. The graph representation for extraction rules overcomes the problems of clausal boundaries, indirect relationships between the predicate and its arguments, and solves the problem of local context. The graph also supports free-word order languages such as Arabic, in case of different word orders are used to describe the same event. The proposed technique for IE depends on graph matching to get benefits of the efficient and well defined algorithms for graph matching, this eliminates the need for parsing the documents, which is very expensive in terms of time and resources.

Keywords: - Lexical Acquisition, Information Extraction, Information Management, Knowledge Management, Document Classification, Document Representation, And Semantic Graphs.

1. Introduction

Information Extraction (IE) is the process of identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship [4]. IE therefore involves the creation of a structured representation of selected information drawn from the text. Another view for IE is the problem of semantic matching between a user-defined template and a piece of information written in natural language [5]. IE can be used as a tool to summarize a text in natural language for specific domain, by extracting the events and relationships between entities involved in the events, then expressing this information in a natural language [2], [8]. Lexicon is a comprehensive knowledge repository where representation and content support many deductive processes such as inheritance, forward/backward reasoning, and constraint propagation [1]. Several studies have been proposed for automating the construction of domain-specific lexicons from three main resources: Machine Readable Dictionaries (MRD), Lexical Knowledge Bases, and application corpora [1]. MRD have been largely used to build computational dictionaries, better suited to be coupled with parsers, or to populate lexical knowledge bases. Many problems depend on the fact that heterogeneous formats of dictionaries pose hard problems to the induction process. Complex extraction processes are required to map the dictionary content to specific data structures [1]. The static nature of dictionaries arises further limitations on using them as a source for lexical Acquisition (LA). In addition, specialized languages, with their own style and phenomena, and with a specific growth rate are very difficult to characterize just relying on dictionary definitions. Many researches and systems are proposed in corpus-driven LA, which use large-scale corpus to study general phenomena in a language; but this technique (corpus-driven LA) requires a large set of training corpora, which is not the normal case. Another approach is to use machine-learning techniques to learn extraction rules from small set of examples annotated by description of each element and the relation between them [6]. Another technique for

LA is building an interactive tool for customizing IE system [4]. In which a user gives an example and the template to be extracted. The system responds by using the existing patterns to create a structural description of the example. It then can interact with the user to extend and generalize the example, both syntactically and semantically. Syntactic generalization can be produced through a set of metarules; semantic generalization can be produced through *isa* hierarchy.

2. Proposed Framework

The objective of this work is to design a lexicon suitable for information extraction from Arabic texts, and to acquire this lexicon automatically for specific domain from set of electronic documents. To achieve this goal we have taken the following actions:

1. Represent the document as well as the domain knowledge.
2. Extract the document and domain knowledge.
3. Test the quality of induced knowledge by performing document classification.
4. Design a lexicon suitable for IE tasks.
5. Design a technique to extract information from text using the domain representation and IE lexicon.
6. Fill that lexicon automatically from the acquired knowledge.

2.1. Document Representation and Classification

The proposed approach for representing document and domain knowledge is to learn the relations between words within the sentence and to compose a word graph for the whole document containing most frequent relations. The same approach is used to compose word graph for the whole domain from the training documents. To ensure that the resulting graph efficiently and sufficiently represents the domain, document classification is then performed using graph-matching algorithm to classify and rate new documents, then the best-induced graph is used for information extraction.

2.2. Document Representation and Feature Selection

It is certainly true that many words in a language have more than one meaning, a property usually called polysemy [9]. There are at least two types of ambiguity [7]: the first is contrastive ambiguity when the word carries two or more distinct and unrelated meanings, the other type of ambiguity involves many senses which are manifestations of the same basic meaning of the word as it occurs in different contexts. These types of ambiguity can be only disambiguated within the context or domain in which the word appears. The idea behind the graph representation is that we search for most representative words in the domain and to find the context in which these words appear. Since using only most important words may be conflicting with similar domains, it is more reliable to match the context in which these words appear. For example, in domain Airplanes' Disasters, most frequent words include 'airplane', 'killed', and 'explosion', but in other close domains like Earthquake Disasters, words like 'killed'), 'explosion' are also frequent, other domains like Airplanes Manufacturers the word 'airplane' will be very frequent. By using the word context, we can detect with high confidence the documents related to the target domain. Our document and domain representation is similar to the network used to measure IE domain complexity of domains regarding to information extraction tasks. Our graph representation also can be used to show word senses if graphs of many domains are combined together. The resulting graph then shows different senses of common words in the combined graph.

2.3. Domain-Model Learning Technique

In this section, we introduce two algorithms to learn the word graph for documents and domains. The first one is suitable for document classification problem; the second one is more suitable for the task of information extraction.

2.3.1. Window Algorithm

The learning algorithm for domain model, as word graph in [3], has two passes; the first pass scans all training documents and performs morphological analysis for each word. The resulting words are stored in *document-word-list* for each document. Each word has a frequency within the document (*Term Frequency TF*) and a frequency within the whole domain (we call it *TDF*). These lists are ordered by *TF* and *TDF*. The second pass scans all documents to determine word relations or graph edges. For each sentence within the document, the top word in *document-word-list* - yet not included in the graph - is searched, if found the *p* preceding words and *f* following words are collected. If the number of surrounding words exceeds *m* minimum number of words per path, an edge is added to the graph between each adjacent pair of words in that path. If the edge already exists in the graph, the edge frequency is increased by one and the average distance is recalculated.

This process is repeated until no words having *TF* more than *mtf (minimum TF)* and *TDF* more than *mtdf (minimum TDF)* are found. If no path discovered in a document, some or all parameters *p (preceding words)*, *f (following words)*, *m (minimum no. of words per path)*, *mtf* and *mtdf* are automatically adjusted to find at least one path in the document. These parameters are recorded for each document. After this pass, the user reviews the output graph for each document and, if needed, changes learning parameter and rebuilds the document graph. This user interaction is an optional step; the process can be continued without it. After all document graphs are built and confirmed by the user, the domain graph is built as the union of all document graphs, edges frequency and average distance are recalculated for the domain graph. Then the lowest frequency (less than certain value) edges are removed from the graph. We called this algorithm as 'Window Algorithm' since we collect certain number of words surrounding important words. This algorithm does not capture all occurrences of word relations or graph edges, since just a word is added to the graph while processing the top of *document-word-list*, it is not searched again even if it is still has high frequency in *document-word-list*. But this technique is sufficient for document classification and has many advantages; first, it reduces number of document scans, secondly, it focuses on the most important words and their relations, and preserves a representative relative frequency at the same time, lastly, it reduces number of features for the document and domain models.

2.3.2. Full-Sentence Algorithm

After testing the graph generated by the 'Window' algorithm, we introduced a new algorithm to build the graph by using the full sentence, not just some words surrounding important ones, since the final goal is information extraction, which need not to miss any information. In this way, only one pass is used to scan a document to build document graph since all words now are considered sequentially regardless their frequency. This method gives a larger graph i.e. larger number of nodes, but including all document information in the graph, which is very important for IE domain.

2.4. Document Classification and Rating

When classifying a new document, the same process for building word graph for the document is performed, except for restricting minimum domain frequency for words, and of course no user interaction.

2.4.1. Bigrams Matching Algorithm

The simplest approach to classify new document given its word graph G_{doc} is to check if it is a proper subgraph of the domain graph $G_{dom,}$ if so then the document is classified as a member of the domain, this approach is called *graph closure*. This approach may filter out some related documents if just one edge (even if it is not significant) in G_{doc} is not in the domain graph G_{dom}. We used another approach that is to get the maximum common subgraph for domain and document graphs and evaluate the resulting subgraph G_{int}. The only deficiency in this algorithm is only single edge matching is performed, that means only pairs of adjacent words in the document are checked against pair of adjacent words in the domain.

2.4.2. Path Matching Algorithm

We developed an algorithm for path matching similar to the above algorithm. After getting the intersection graph G_{int}, we trace its edges to get all paths of adjacent edges for each sentence. For each path *p*, its length *l(p)* and frequency *freq(p)* are calculated. We found that algorithm gives better document rating than the bigrams matching algorithm. We achieved 0.95 for F and Accuracy measures for classification by both algorithms, complete experiment results can be found in [3].

2.5. Semantic Lexicon for Information Extraction

We proposed a design for generative lexicon suitable for information extraction tasks; the lexicon is composed from two parts: terms and events. The relation between terms and events is many to many. The term may have many senses and holds the attributes needed to describe it for each sense.

2.5.1. Proposed Extension for the Arabic Lexicon

We propose an extension to the lexicon that is to add references to one or more semantic classes, within hierarchical classification to represent the world model, for both roots and non-derived nouns; verbs and derived nouns will inherit their semantics from the root.

2.5.2. Proposed IE Lexicon Design

Our proposed lexicon for IE consists of two hierarchical components:

Terms

Each term in the lexicon has the following structures:

1. Quale Structure: which denote to the different meanings (types) of the term.
2. Argument Structure: defines the attributes of the term in case of nouns, and the syntactical arguments (subject, object…) in case of events (verb). Associated with each argument a pattern to describe how this argument is realized in the syntax and with which meaning (quale) of the term.
3. Event Structure: specifies all events associated with the term in case of nouns. Again, associated with each event a pattern to describe how this event is realized in syntax and with which arguments and qualia.

As an example, the term plain (in Arabic) may have the following structures:

***Qualia*: x :** *Aerial Transportation mean*
Y: *Adjective of flying object*
Z: *fight mean*
***Arguments*:**
USAGE **– Pattern:** *{Travelers,*
Shipping, Exploration}
Owner – **Pattern: { X** *is owned by <owner>}*
Nationality – **Pattern:***{X <Nationality>}*
…
***Events*:**
Landing **– Pattern***:{ X landed <status>*
at <place>}
Falling – **Pattern:** *{ Y X felled in <place> }*
…

Domains

Domains are defined in a multiple inheritance hierarchy. Each domain has the following structures:

1. Event Order: a domain consists of many events with some alternative orders. We used "<" to denote "because of" or "before" relation between two events and "=<" to denote "while" relation.
2. Terms: finite number of designating terms can describe a certain domain, so the domain in the IE lexicon will be associated with its most frequent terms. 'Exclude' flag is used to specify excluded arguments or events from the term.

Consider the following example:
Domain: Airplane *disasters*
Event Order: {killing < crash}, {crash, falling,
take off}, {crash < collision =< take off}
...
Terms:
(Plane ((Quale X: Transportation mean)
(Events (Manufacturing , Exclude =True)
, (Experiment, Exclude =True)
, (design, Exclude =True)))
, take off((Quale E: Object is raised over
the land))).

2.6. Semantic Representation for IE Domains

The semantic lexicons have the same importance as full parsing for Scenario Template (events) task and they are very useful in Template Elements (entity attributes) and Template Relations (entity relationships) as well. Since the proposed IE semantic lexicon captures the syntactic and semantic patterns with relationships between terms, we introduce a semantic representation for the document and domain knowledge suitable for information extraction, that is: instead of using words as the graph nodes, we use the semantic class of the word as the graph node. Using semantic class of the word instead of the word itself generates smaller graphs and highlights the important semantic classes within the domain more than word graph. To build the semantic graph for IE domains from corpora, the general-purpose lexicon is used to analyze each word in order to get its syntactic and semantic properties. Graph definitions are used but the node is the semantic class of the word instead of the word itself. Each document in the corpora is analyzed to build its semantic graph, and then the domain graph is the union of all document graphs with edge frequency is the summation of frequencies of that edge in all documents.

Path	Freq.	Nodes No.
<Nationality ><Transportation Mean>	48	2
<confirmation article><Transportation Mean>	35	2
<falling><transportation mean>	27	2
<Nationality><confirmation article>	26	2
…	…	…

2.7. IE Lexicon Acquisition from Corpora

We propose a semantic graph as an extraction pattern for each event (and its entities) of interest within each domain. The end user has to specify the events and semantic classes of the entities involved in each event as well as the attributes of interest for each event and entity. The graph representation for extraction patterns has many advantages in addressing the problems of other pattern representations: (Indirect relationships, Clausal boundaries limitation, Free word order problem, Lack of context problem, and The problem of large number of patterns). Typically, an event has some attributes such as location, date, and time, and has some entities acting together to perform that event, each entity has some attributes, and one entity may be involved in many events within one domain. The challenge here is how to find the borders of the event/entity and its attributes within the domain graph. We propose an approach that is to start with finding the nodes in the graph representing the event/entity attributes, then find the most important paths that link those nodes together to form one disconnected graph. Adding one path at a time then use the induced graph so far to extract information from the testing documents until maximum performance is reached. Due to the large search space, and the expensive evaluation function for current state, we use hill-climbing technique to find the best move to a new state. But this technique suffers from local maxima and plateau problems, to avoid those problems we use the strategy of finding a distant point whenever we stuck at a local maxima or plateau and continue the search normally from that point, all discovered local maxima are recorded to avoid getting trapped in them again. Following are the most frequent edges (with frequency more than 5) in the semantic graph of event fallen in the domain Airplane disasters

From Node	To Node	Frequency
<fallen>	<transportation mean>	37
<adverb>	<place>	34
<beginning>	<month>	18
<killing>	<human>	17
…	…	…

2.8. Graph Matching for Information Extraction

The proposed technique for IE depends on graph matching to get benefits of the efficient and well-defined algorithms for graph matching; this also eliminates the need for parsing the documents. In addition, in most current IE systems, the parsing is done locally at the clause level, this causes some information described over many clauses and sentences are hard to extract. The graph covers this shortcoming since it represents the whole relations within the domain regardless the sentence boundaries. The system captures the knowledge of a new domain by generating domain semantic graph from the training corpora supplied by the user, then the user provides a set of the events and entities of interest with their attributes, the system generates semantic graphs for them. All semantic graphs are stored in the IE lexicon for the underlying domain to be used in extraction. The extraction process for certain domain from new set of documents starts by selecting the relevant documents to the domain. Relevant documents are retrieved from the new documents set using the document classification technique. Then, the extraction process starts on each document.

3. Conclusions

Document and domain graph representations for both document classification and information extraction are proposed as well as algorithms to build those graphs, with a new technique for document classification based on the graph representation for both documents and domains. Representing the domain model as a graph leads to document classification with very high precision and recall, as well as reduced the number of used features, which improves the system performance. Two algorithms for building the document and domain graphs are proposed: Window and Full-Sentence algorithms, the former gives smaller graph suitable for document classification, but the second gives larger graph more suitable for IE. Two algorithms for classification are proposed: bigrams-matching and path-matching algorithms; path-matching algorithm gives better document rating than bigrams-matching algorithm. We proposed semantic graph representation for IE domains using semantic class of the word instead of the word itself, which generates smaller graphs and highlights the important semantic classes within the domain more than word graph. We introduced an unsupervised learning algorithm to build semantic graphs from corpora without any annotation or a pre-classified corpus with relevance judgments, or any feedback or intervention from the user. The algorithm uses general-purpose semantic lexicon to perform morphological analysis and extract syntactic and semantic information for words in the domain. The resulting semantic graph represents the domain knowledge and is used to induce extraction patterns for events and other terms within a domain. The graph also supports free-word order languages such as Arabic, in case of different word orders are used to describe the same event. We also used the same representation – semantic graph – for extraction rules instead of the common regular-expression-like rules. The graph representation for extraction patterns has many advantages in addressing the problems of other pattern representations such as clausal boundaries, indirect relationships between the predicate and its arguments, and solves the problem of local context. A language-independent lexicon design for IE is proposed to be attached with a general-purpose semantic lexicon, as well as an algorithm for acquiring extraction semantic graphs for domain events and entities with their attributes from the domain graph to fill the IE lexicon. We introduced an algorithm for extracting information from free text using well-defined and efficient graph matching techniques. We represent the new document using the same representation as the domain, then match document graph with event and entities graphs stored in the lexicon to extract the desired information. As a future work, we suggest integrating the acquired IE lexicon from Arabic corpora with an Arabic IE system to test the lexicon validity and to measure the performance of the introduced acquisition and extraction techniques in this paper.

References:

[1] Basili, R., and Pazienza M. T. 1997. *Lexical Acquisition for Information Extraction*. In Maria Teresa Pazienza (Ed), *Information Extraction: A multidisciplinary Approach to an Emerging Information Technology*; international summer school / SCIE-97, Frascti, Italy, July 14 – 18, 1997.

[2] Gaizauskas R., Humphreyes K., Azzam S., and Wilks Y. 1997. *Conceptions vs. Lexicons: An Architecture for Multilingual Information Extraction*. In Maria Teresa Pazienza (Ed), international summer school / SCIE-97, Frascti, Italy, July 14 – 18, 1997.

[3] Gheith, Mervat., Aboul-Ela, Magdy, and Arafa, Waleed. 2002. *Learning Word Graph Representation for Document Classification*. Proceedings of the 27th Conference for Computer Science, Statistics and Operation Research, Egyptian Computer Society; Cairo, 13-18 April, 2002

[4] Grishman, R. 1998. *Information Extraction and Speech Recognition*. http://www.nist.gov/speech/proc/darpa98/html/sdr10/sdr10.htm. Site visited at May 11, 1999.

[5] Guarino N. 1997. *Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration*. In Maria Teresa Pazienza (Ed), international summer school / SCIE-97, Frascti, Italy, July 14 – 18, 1997.

[6] Muslea, I. 1999. *Extraction Patterns for Information Extraction Tasks: A Survey*. AAAI'99 Workshop on Machine Learning for Information Extraction, July 19, 1999, Orlando Florida.

[7] Weinreich, U. 1964. *Webster's Third: A Critique of its Semantics*. International Journal of American Linguistics 30, pp. 405-409.

[8] White, M.; Korelsky, T.; Cardie, C.; Ng, V.; Pierce, D.; Wagstaff, K. 2001. *Multi-document Summarization via Information Extraction*. Proceedings of HLT 2001, First International Conference on Human Language Technology Research, J. Allan, ed., Morgan Kaufmann, San Francisco, 2001.

[9] Pustejovsky, J. 1998. *The Generative Lexicon*. The MIT press Cambridge, Massachusetts, London, England.