EXTRACTING ARABIC COLLOCATIONS BASED ON JAPE RULES

Soraya zaidi *, Mohamed-Tayeb Laskri*, Ahmed Abdelali**

*Department of computer science Badji Mokhtar University, BP 12 Annaba 23000, Algeria
Soraya.zaidi@univ-annaba.org, laskri@univ-annaba.org
**Research Administration, MSC 3Res New Mexico State University, USA.
aabdelal@nmsu.edu

ABSTRACT

The massive amount of digital information available in all disciplines has generated a critical need to organize and structure their content. Among the existing tools for languages such as English or French can easily be adapted to Arabic language. In some cases a simple configuration is sufficient while in other cases significant modifications must be made to obtain acceptable results. We present in this paper a rule-based method for extracting collocations in Arabic language using Gate1 (General Architecture for Text Engineering). We use the extracted collocations as domain terms to build Arabic text-based ontologies. We validated the approach using The Crescent Quranic Corpus in order to build automatically the Quran ontology.

Keywords: Collocations extraction, Arabic language, Jape, NLP, GATE.

1. INTRODUCTION

Arabic is a Semitic language spoken by over 300 million speakers in over 22 countries. Until the advent of Islam in the seventh century, Arabic was essentially an oral language. The Quran, Islam's holy book, was revealed to the Prophet Muhammad (PBUH) in Arabic, giving the language a great religious significance and was the main motive that carried this language beyond its native geographic space. Arabic has 28 letters and is written from right to left. Arabic language processing is considered complex because of structural and morphological characteristics, such as inflection, polysemy, and irregular forms of words.

There are a variety of tools to extract terms in languages such as French; English, some of them can easily be adapted with some minor modifications for the Arabic language. GATE for example may be used in Arabic named entities extraction. Named entities represent person names, organizations, places, money, time or other numerical expressions. EXIT [17] can be used to extract collocations but it needs a special Arabic tagger. Currently we are working on adapting GATE to extract Arabic collocations such as Noun-Noun, Noun-Adjective, verb-Noun, Noun-Preposition-Noun, etc... We extract collocations from Arabic corpora by creating new transducer and using Jape rule for every pattern.

In the second section of this paper, we present some of related works in section three, we introduce the Arabic information extraction. We discuss in Section four the main modules of GATE and how it can be used to extract Arabic collocations, later in Section five we provide an assessment and evaluation in term of precision, recall and F.measure. In conclusion we give a future and some perspectives for our work.

2. RELATED WORKS

In the followings we present some of the widely used tools for extracting terms despite of the language [9].

2.1. ACABIT :
ACABIT is a terms extraction tool from French tagged corpora using linguistic approach, it was developed by Beatrice Daille in IBM company [12].

2.2. LEXTER :
LEXTER (Logiciel d’EXtraction de Terminologie) was developed by Didier Bourigault in EDF (Electricité De France) to help in data mining terminology extraction. LEXTER extracts candidate terms from a tagged and disambiguating corpus [13].

2.3. FASTER :
FASTER is a robust parser for recognition terms in corpus. FASTER uses a set of predefined rules to identify terms [14].

2.4. XTract :
Xtract is not specifically dedicated to the terminology extraction but it has its place in this description because it served as inspiration to some acquisition terminology works based on statistical techniques. This tool is a collocation extractor[15].

2.5. ANA :
ANA is a terms extracting tool that extracts candidate terms without any linguistic analysis. The terms are recognized through approximate equalities between words and observed repetition of patterns [16].

2.6. EXIT :
EXIT (Extraction Iterative de Terminologie) is a terminology extraction system from the specialized

1 http://gate.ac.uk
domain corpora. EXIT is semi-automatic and should be used by domain expert users, who must have a great knowledge of used concepts in analyzed texts in order to recognize the relevant collocations among those extracted by the software. EXIT is based on a statistical and linguistic approach[17].

2.7. SYNTAX:
SYNTAX is a syntactic parser. There are currently French and English versions. After syntactic parsing of the corpus SYNTAX builds a network of words and phrases (verbal, nominal, adjectival) called a "Terminology network" in which each phrase(or syntagm) is linked to its head and its expansions. The network elements (words and phrases) are called "candidate terms [18].

3. ARABIC INFORMATION EXTRACTION
3.1. GATE TOOL
GATE is an infrastructure for developing and deploying software components that process human language. GATE helps scientists and developers in three ways:

1) By specifying an architecture, or organizational structure, for language processing software;

2) By providing a framework, or class library, that implements the architecture and can be used to embed language processing capabilities in diverse applications;

3) By providing a development environment built on top of the framework made up of convenient graphical tools for developing components [10].

The GATE architecture is based on components: reusable chunks of software with well-defined interfaces that may be deployed in a variety of contexts [11].

- CREOLE (Collection of REusable Objects for Language Engineering). We can define three types of components:
  - **Language resources** (LRs) : This is a number of linguistic data such as documents, Corpora, Ontologies...
  - **Processing Resources** (PRs): These are programs or algorithms which will do some sort processing on text i.e. Tokenising or dictionary lookup, parsing etc.
  - **Visual Resources** (VRs): These are components for graphical user interface and allow viewing and editing of other types of resources.

- **ANNIE** (A Nearly-New Information Extraction system): GATE was originally developed in the context of Information Extraction, Annie is the main component. ANNIE components form a pipeline including a parser, a gazetteer, tokenizers (with disambiguation), a tagger, orthomatcher [2].

GATE-based systems are in use at several VoC suppliers, including a company that analyses customer feedback from some of the largest transportation organisations in the UK, and a New York customer sentiment start-up. Gate is used in several projects, cancer research, for web mining of personal data; several companies have GATE-based mining of job adverts and CVs. Text2Onto uses Gate architecture for text pre-processing [19].

- **JAPE Language**
JAPE (Java Annotation Pattern Engine) provides finite state transduction over annotations based on regular expressions. JAPE is a version of CPSL (Common Pattern Specification Language) [3]. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consists of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements [4].

3.2. EXTRACTING ARABIC NE WITH GATE

We experimented with Gate using two different Arabic corpora [20], Corpus of Contemporary Arabic (CCA) developed by Latifa Al-Sulaiti at Leeds University [5] which consists of over 843,000 words in 416 files covering a wide range of categories. Formatted as XML documents, the second corpus was The Crescent Quranic Corpus, by Kais Dukes [6] which consisted of Quran text tagged per word, verse and chapter; it contains as well additional information about morphology and POS.

The processing resources of the Arabic language that we used are:

- Arabic gazetteer: The gazetteer lists are plain text files with one entry per line, each list represents a set of names such as names of cities, organizations, locations etc. This type of gazetteer is built manually.
- Arabic inferred gazetteer: This gazetteer is deducted automatically from training data.
- Arabic main grammar: Allows you to use files containing the various rules.
- Arabic tokeniser: Segments the Arabic text in simple tokens such as numbers, punctuation and words of different types.
- Arabic orthomatcher: To solve the problem of coreference.

After selecting PRs, for a pipeline, the application is running and the result is displayed as text annotations. Of course it is possible to correct or add annotations manually by selecting the word to annotate and assigning the appropriate annotation.

4. ADAPTING GATE TO EXTRACT ARABIC COLLOCATIONS

4.1. ARCHITECTURE

GATE can support the following document formats: Plain text, HTML, SGML, XML, RTF, Email, PDF, Microsoft Word. By default GATE will try to identify the type of the document, then strip and convert any markups into Gate’s annotation format. To disable this process, set the markupsAware parameter on the document to false.

Gate was built mainly for named entities extraction; our idea is to adapt it to collocations extraction from Arabic tagged corpora using predefined patterns. Extracting terms automatically from corpora is to identify elements carrying a key concept in the specified domain. Terms may be simple or compound words form as Noun-Noun: “ Verb-Noun: “، Noun-Adjective:”، Noun-preposition-Noun:” etc. After choosing the pattern if we want to use, for example, Noun-Adjective, we write the appropriate Jape rule, then create a new NE ANNIE transducer, the rule is passed as parameter to the transducer [2], collocations in Arabic text will be detected, displayed and the annotated document can be saved in datastore with the new tag or label which could be exploited for other tasks or used as an input for another experiment.

We have experimented with the transducer on the Crescent Quranic corpus [6], the aim was to get a list of terms for a given domain and use the resulted list after filtering as concepts for the automatic construction of ontologies. The output can also serve as a basis for developing terminology or a thesaurus. Jape rules define new annotations used to detect collocations such as Name-Adjective Name-Name, Verb-Name etc.
Each Jape rule grammar has one of 5 possible control styles: 'brill', 'all', 'first', 'once' and 'appelt'. This is specified at the beginning of the grammar. The Brill style means that when more than one rule matches the same region of the document, they are all fired. The result of this is that a segment of text could be allocated more than one entity type, and that no priority ordering is necessary.

On the LHS, each pattern is enclosed in a set of round brackets and has a unique label; on the RHS, each label is associated with an action. In this example, the annotation is labelled “SomeLabel” and is given the new annotation N_ADJ.

The creation of new transducers with the previous settings, will allow identifying collocations according to specified syntactic patterns. A validation by human expert in the domain is carried after. This is consists of accepting or rejecting collocations displayed, because it is possible to get words that validate the pattern but the obtained set is not considered as a relevant collocation.

The ‘all’ style is similar to Brill, in that it will also execute all matching rules, but the matching will continue from the next offset to the current one. With the ‘first’ style, a rule fires for the first match that’s found. This makes it inappropriate for rules that end in '+' or '?' or '*'. Once a match is found the rule is fired; it does not attempt to get a longer match (as the other two styles do).

With the ‘once’ style, once a rule has fired, the whole JAPE phase exits after the first match. With the appelt style, only one rule can be fired for the same region of text, according to a set of priority rules. Priority operates in the following way.

1. From all the rules that match a region of the document starting at some point X, the one which matches the longest region is fired.
2. If more than one rule matches the same region, the one with the highest priority is fired.
3. If there is more than one rule with the same priority, the one defined earlier in the grammar is fired [3].

The previous rule in Figure 3 allows recognizing words in a text with a *Noun* tag followed by *adjective* tag, to give out the collocation consisting of N-ADJ, similarly we can write additional rules to recognize other types collocations.

5. RESULT AND EVALUATION

The aim of our work is to extract collocations from Quranic corpus. After validation, a list of relevant collocations-those are considered as terms of the domain- will be drawn up, in order to use them in automatic building of domain ontologies. After running application with new parameters we can display extracted collocations as shown below.
The collocations extracted as “العربية” “القرآن الكريم” “العربية” can be displayed or saved in datastore for future use.

A vital part of any language engineering application is the evaluation of its performance, in order to evaluate the system performance, we used traditional measures of Precision and Recall (and then F.measure) on our corpus. AnnotationDiff tool [21] enables two sets of annotations on a document to be compared, in order to compare a system-annotated text with a reference (hand-annotated) text. We do on Noun-Adjective annotation. We annotated manually the same documents which are annotated using Jape rules with Gate, then we used AnnotationDiff to calculate Precision, Recall and F.Measure. we obtained 1 to Recall and 0.5 in precision then F.measure gives 0.66 [8]. We know that the Recall is inversely proportional to Precision, then if we want to improve the first we reduce the second and vice-versa. The low precision is mainly due to the ambiguity and the fact that we did not use all features when writing rules that could vastly improve the annotations, that what we are doing now.

6. CONCLUSION AND PERSPECTIVES

We have presented in this paper the problem of extracting collocations from Arabic texts. First we tested Gate on Arabic corpus to extract named entities, using its own resources as Creole, Annie, Owlim which is an integrated editor of ontologies and can be used to populate ontologies with instances, after we used Jape to write new rules in order to extract Arabic collocations. These collocations will serve as terms after validation to build automatically Arabic text-based ontologies. We tried this linguistic method on Crescent Quranic Corpus, our goal is to construct Quranic ontology, work is still ongoing, currently we work on relation extraction using syntactic patterns, our objective is to have a semantic network with extracted terms and relations. We aim to use the resulted ontology as platform to improve information retrieval on the web in this domain; it can be also used to improve machine translation and automatic indexing.

REFERENCES


