Genetic algorithm for extracting relations between named entities

Ines Boujelben, Salma Jamoussi and Abdelmajid Ben Hamadou

Miracl, University of Sfax – Tunisia
Boujelben_ines@yahoo.fr, jamoussi@gmail.com, adelmajid.benhamadou@isimsf.rnu.tn

Abstract

In this paper, we tackle the problem of extracting relations that hold between Arabic named entities. The main objective of our work is to automatically extract interesting rules basing on genetic algorithms. We tend firstly to annotate our training corpus using a set of linguistic tools. Then, a set of rules are generated using association rule mining methods. And finally, a genetic process is applied to discover the more interesting rules from a given data set. The experimental results prove the effectiveness of our process in discovering the most interesting rules.

Keywords: genetic algorithm, relation, named entity, rule mining, interesting rule.

1. Introduction

Extracting useful information from texts presents an important research area. Among these areas, the named entities (NEs) recognition is seen as a crucial task towards semantic analysis. Nevertheless, it presents only the first step for natural language processing. To go beyond the extraction of NEs, the detection of relations involving these entities is required for more structured model of text understanding. This kind of information presents the task of discovering useful relationship between two entities from text contents. It has received a great deal of attention since it is used in many information retrieval like question-answering, automatic summarization and web mining. In this paper, we are based on GA to extract the more significant rules for relation recognition. In fact, GA has been successfully applied in many search, optimization, and machine learning problems. It was developed by (Holland, 1970) and incorporates Darwinian evolutionary theory with sexual reproduction.

The remainder of this paper is structured as follows: First, we provide an overview of works related to relation extraction between NEs. Then, we discuss the motivation of relation extraction for Arabic language. In the next section, we describe our proposed method based on GA (Holland, 1970). Thereafter, an evaluation and analysis of results are conducted to assess the quality of our process.

2. Related work

Many research works have been already performed. They can be classified into two broad categories: rule based and learning-based approach. The rule based approach is focused on manual linguistic patterns, notably (Ben Hamadou et al., 2010). In ML approaches, there has been few works focused on unsupervised approaches that conduct to extract strings of words between NEs in a large amount of non annotated text. In this context, (Hasegawa et al., 2004) elaborated a clustering of NE pairs according to the similarity of context words. Another work here was done by (Zhang et al., 2004) in which they are based on computing the similarity parse trees. The resulting relation of this approach may not be easy to match the relations needed for particular knowledge. Additionally, such method requires a high frequency of NE pairs, which is not the case of relations expressed in Arabic texts.

Some other works have a tendency to use supervised learning patterns in which a set of patterns can be learned from labeled training examples. For instance, (Kramdi et al., 2009) are based on the learning and the selection of patterns using learning pattern algorithm LP^2. After the automatic generation of rules using learning algorithms, (Boujelben et al., 2013) proposed four selection levels. Through these levels, they used filtering and enrichment technique in order to extract the best rules. Despite its low recall, this method achieved satisfactory results in term of the precision.

In this paper, the problem of coverage and precision of rules is tackled through the integration of the GA to enhance the rules generated by Apriori (Agrawal et al., 1993) and C4.5 (Quinlan et al., 1993) learning algorithms basing only on morphological, semantic and numeric features.

3. Motivation

As we know, the extraction of relation between NE has been tackled by many research works in texts written in European languages (English, French, and Italian). As far as we know, there are few works that have been done in the Arabic language. We mention (Ben Hamadou et al., 2010) who are limited only to the functional relation between Person and Organization entities (PERS_ORG) basing on handcrafted extraction patterns transformed into NooJ transducers. In an attempt to cover other kind of NE pairs, using the rule based approach, (Boujelben et al., 2012) have introduced also a rule based method to treat the relation that can be holding between the NE pairs (PERS_PERS, PERS_LOC, PERS_ORG, LOC_LOC and ORG_LOC). For the reason that such approaches need a tangible effort to write patterns and in a tendency to automate this method, (Boujelben et al., 2013) proposed supervised learning rules to extract relations between various pairs of Arabic NEs. Through this paper, we aim to improve the overall coverage of this process using genetic algorithm.

Besides that, the Arabic language suffers from the lack of available Arabic resources like morphological tagger and annotated corpora. In fact, many available corpora are neither annotated with NEs nor include sufficient number

of NEs which are related in order to be used for the machine learning algorithm. Moreover, the electronic texts are not vowled. As a consequence, we were faced to the problem of determining the correct morphologic category of words. For example, the word (كتب) when it is not vowled, it can be presented as a verb that means (write) or noun (book) which can prevent our process of relation extraction. The Arabic language is characterized by its complexity and agglutinative[1] words. Therefore, researchers tend to spend tangible effort to annotate and verify our own Arabic resources.

In the relation extraction task, we can have two NEs in the same sentence but they are not semantically related. To tackle this problem, (Wang et al., 2011) proposed heuristics in which they remove the relations that contain reporting verbs such as (talk, said…). Also, they considered only the relation that contains only one verb between NEs excluding the case that contains an auxiliary verb. But, a relation between Arabic NEs can be expressed through noun, punctuation, adjectives… And, the reporting verb can indicate a relation; For example, the sentence "Michel talk to Stephanie ", presents a relation expressed through the word "talk" between person NEs "Michel" et "Stephanie". In our work, this problem is tackled through the segmentation of sentences into clauses. Thanks to this tool, we ensure that every two NEs presented in one clause are certainly related.

Given a relation name, labeled examples and a training corpus, traditional relation extraction systems output instances of the given relation recognized in the corpus. For our method, relation names are not known in advance. We consider the word position that explicit the relation as an output class. Unlike other works (Delloye, 2008) and (Wang et al, 2011) who are limited on specific relation types, this trigger word can be located before the first NE, after the second NE or between these two entities. Thereby, we can extract an infinite number of semantic relations expressed through these trigger words. In a subsequent step, we will elaborate a semantic classification of these triggers. So, we extract not only the semantic relation class, but also the instance of this relation which provide more precise relation.

4. Proposed method

The relation extraction process is performed in two steps: the generation of rules using learning methods and the discovering of the most interesting rules using GA.

4.1. Rules mining

A rule is defined as a conditional statement that can easily be understood by humans and easily used within a database to identify a set of records. In our case, a GA is applied over rules fetched from mining rules algorithms.

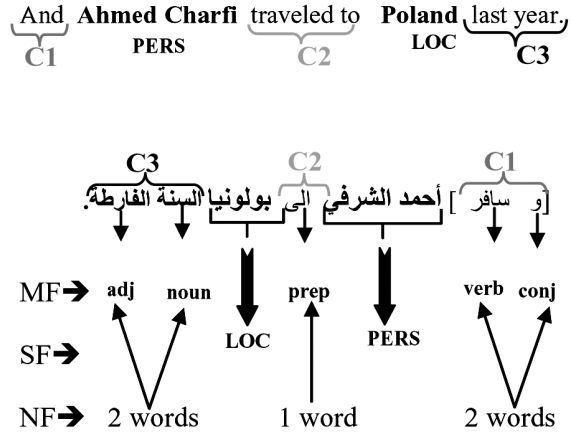
In fact, we investigated the unsupervised algorithm Apriori (Agrawal, 1993) to generate class association rules and the C4.5 algorithm (Quinlan, 1993). Indeed, this algorithm generates classifiers expressed as decision trees Because, it can learn rules from training samples and it can match other instances not covered by the training data, we tend to use this algorithm. The resulting tree obtained by C4.5 can be converted into a set of rules having the form of: "if Attribute1 and Attribute2 and Attribute3 then class X".

Our training data is composed of texts collected from electronic and journalistic articles in Arabic. These sentences contain 2000 NEs, where only 1204 NEs are related. As a consequence, our dataset is composed of 1102 instances. We are interested only on the following pairs of NE: PERS_LOC, PERS_PERS, PERS_ORG, ORG_LOC and LOC_LOC.

From our training corpus, we extract, firstly, clauses that contain only two NEs. In fact, a clause is composed of a set of words that contains a subject and predicate. It can take various categories (noun clauses, relative clause or verbal clause). Hence, it can be presented also as a sentence. This extraction required an Arabic clause splitter as well as Arabic NEs recognition tools. Then, these clauses are automatically annotated using a morphological parser.

Given these annotated clauses, we abstract the same linguistic features used in (Boujelben et al., 2013) such as NEs tag, part of speech of words surrounding the NEs and the number of words before, after and between the NEs.

Let's see the following example:



ML= Morphologic features: The POS tag of context words.
SF=Semantic features: The semantic type of NE (PERS, LOC, ORG).
*NF=Numeric features: number of words of each context (**C1**: words after the first NE, **C2**: words between NEs, **C3:** words before the second NE).*

The application of learning algorithms produces an important number of rules which can be in some sense interesting or not. For this reason, we envisage to apply genetic operations to these rules in order to cover further instances and to enhance the precision.

4.2. Discovering interesting rules using genetic process

Seeing that genetic algorithm has been successfully applied in many searches, optimizations, and ML problems, in recent years numerous works have been carried out using the evolutionary algorithm for mining rules. This is why, we adopt this process to automatically extract interesting rules.

[1] In Arabic language, particles such as pronoun and preposition can be attached to the word.

In our genetic process, we use the Michigan approach in which each chromosome represents a separate rule. The main idea is to progressively improve the quality of initial rules by constructing new fitter rules until either rules of high quality are found or no further improvements.
We first apply a filtering module in order to initialize the population for our genetic algorithm. The selected rules will be then undergoing the reproduction methods which are the crossover and mutation.
For the crossover operator, we tend to conserve the useful information in two given rules. The crossover children are created by combining the elements of their parent rules. In fact, each pair of rules disposing the same class is randomly chosen and it is crossed over to reproduce two descendant rules.
It consists in selecting a random position in each parent rule. Then, we swap all the genes after that point.
This genetic operator is demonstrated in the following figure:

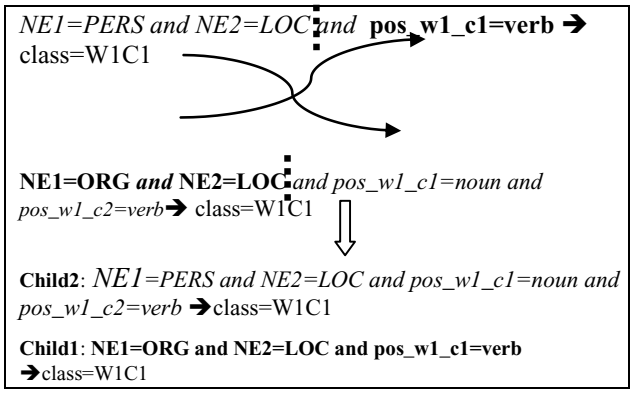


Fig. 1: Illustration of the single point crossover operator

Whenever rules are chosen from the population and are crossed-over, our GA verifies the mutation probability P_{mut}. The dynamic P_{mut} favors to mutate the high quality rules (R) as follows.

$$P_{mut}(R) = (1\text{-confidence}(R))/10$$

Then, each new rule that is being chosen in term of probability will mutate. Unlike ASGARD (Jourdan et al., 2002) who use value and attribute mutation, our genetic operator consists in excluding one constraint of a given rule to obtain more generic rules that cover more instances. Hence, we prefer to conserve the useful information offered by our rules since they are generated from learning algorithms and are already filtered in terms of confidence[2] and length. Thus, for each rule, we remove one item (the attribute and its value) and keep the rest of the rule to obtain the derived rule. This process is applied for each item.
Hence, we obtain a big number of derived rules which is equal to the length of the each parent rule. The mutation operator is illustrated in the following figure.

NE1=PERS and NE2=LOC and pos_w1_c1=verb and pos_w1_c2=noun and pos_w2_c2=prep➔ class=W1C2

child1: NE2=LOC and pos_w1_c1=verb and pos_w1_c2=noun and pos_w2_c2=prep➔ class=W1C2
child2: NE1=PERS and pos_w1_c1=verb and pos_w1_c2=noun and pos_w2_c2=prep➔ class=W1C2
child3: NE1=PERS et NE2=LOC and pos_w1_c2=noun and pos_M2_C2=prep➔ class=W1C2
child4: NE1=PERS and NE2=LOC and pos_w1_c1=verb and pos_M2_C2=prep➔ class= W1C2
child5: NE1=PERS and NE2=LOC and pos_w1_c2=noun and pos_w1_c1=verb➔ class= W1C2

Fig. 2: Illustration of the single point crossover operator

As a result, the coverage of our system becomes better while the precision is reduced. Our main objective is to obtain the best compromise between rules number, precision and recall. We need to find generic rules with an acceptable precision.
To address this issue, we pass to the replacement step in which we reinsert these descendant rules into the initial population to create the new current population.
In this phase of creating new generation, the GA usually selects the best rules as parents. Moreover, the rules that have higher quality are chosen as elite. These elite individuals are passed to the next population. For the rules derived from a top rule through the mutation operator, we compare each target rule with its offspring in order to satisfy two main assumptions:

- Each derived rule that holds with a confidence value more than a specified threshold and gets support[3] higher than the support of the top rule will be selected.
- In the case that all derived rules have confidence values below the threshold value, we will conserve only the target rule and eliminate all derived rules.

Finally, the last obtained rules will be sorted according to the confidence and support values to only keep the best ones.
GA runs to generate solutions for successive generations until either interesting rules are found until stagnation of the population evaluation or a fixed maximum number of generations have been reached. That means, if we have no change in two or more consecutive generations, we can stop the GA process. As a result, GA generates a population at last, with high quality rules.

5. Experiments and evaluation

For the evaluation, we use ANERCorp corpus (Benajiba et al., 2007) as a test corpus. This corpus is composed of more than 316 articles, containing more than de 150,000

[2] It shows how frequently the rule head occurs among all the groups containing the rule body.

[3] It presents the number of instances in which the rule is applicable (either correct or false).

words and 3206 labeled NEs. We have only 840 NEs that are related in a clause. We are interested only on the NEs (PERS, ORG and LOC).
Firstly, we extract only clauses composed of two NEs using the clauses splitter of (Keskes et al., 2012) and the morphological analysis is accomplished using the Arabic resources of (Mesfar, 2008). As a result, we obtained 420 instances in our test database.
Two experiments have been conducted to evaluate the efficiency of our system. The first experiment concerns the evaluation of our proposed GA when applied to:

- (E1) rules randomly generated: An initial population is created consisting of randomly generated rules. Indeed, the number of attributes of one rule is chosen in a random way between 3 and the maximum number of attributes of the database. Also, the rule class is chosen randomly. Each rule must cover at least a record of the database. We apply our GA to obtain the more quality rules departing from randomly rules.
- (E2) rules produced by Apriori which aims to find all the rules existing in the database that satisfy some minimum support (minSup) and minimum confidence (minConf) constraints. For our case, we take (minConf=0.6 and minSup=2).
- (E3) rules produced by C4.5.
- (E4) rules generated by the combination of these two algorithms Apriori and C4.5.

The evaluation metrics used are the precision[4] (P), recall[5] (R) and F_score[6] (F).
For the GA parameter setting, the mutation probability is taken as 0.04, the crossover probability is taken as 1, the initial population size is 190 and maximum number of generations is 1000.

	Before GA	After GA
	(P | R | F) %	(P | R | F) %
E1		(51,2 | 54 | 52,6)
E2	(99 | 25,6 | 45)	(60,2 | 51 | 55,2)
E3	(13|10,6 |12,3)	(28 | 30,2 | 29,1)
E4	(55,8|42|52,8)	(74,1|59,6| 66,1)

Table. 1: Evaluation of GA for various rules sources

In the second experimentation, we compare our process basing on genetic algorithm with the results obtained by (Boujelben et al., 2013) when applied to ANERCorp corpus.

	P	R	F
(Boujelben et al.,2013)	62,03%	54,52%	58,03%
Our method	74,1%	59,6%	66.1%

Table. 2: Comparative evaluation

The results shown above (Table1) demonstrate that the genetic process applied to the rules generated from the combination of Apriori and decision tree algorithms obtain the best results in terms of precision and recall. It is understandable since we have various rules produced by two different mining algorithms: Precise association rules produced by Apriori and classification rules generated by C4.5.
The table2 demonstrates that our GA based method obtain best results in terms of both precision and recall. It shows 8% increase.
In practice, we distinguish quite errors and noise ones. The quite errors are caused by the morphological ambiguity of our Arabic language. Indeed, in some cases, a NE can be analyzed as a part of speech of a given word. For example the Arabic NE ([illegible]) can be [illegible] Since a primordial argument of relation extraction is omitted caused by the non recognition of this word as a NE, the concerned clause will be not taken into account to be treated.
We mention also the influence of NE recognition ambiguity. For example, ([illegible]) could be either identified as a name of a person PERS or a name of a country LOC. As a consequence, noise errors will produced when applying a rule to the associated instance. Also, in the same context, we cite the influence of the morphological parser of words on the performance of our process.
Then, given that we are based on supervised learning method which required annotated database, it seems crucial to increase the number of annotated instances.
And finally, through the integration of two mining algorithms with GA, we achieved satisfactory results better than the work done in (Boujelben et al., 2013). The great precision of obtained rules is largely related to the performance of our genetic operators (crossover and mutation) as well as the replacement step that aims to choose in each generation the most interesting rules to replace the less quality rules. As shown in table1, our genetic process increased the F_score for each rules, either generated by Apriori and C4.5 separately or produced by the combination of them.

6. Conclusion

Among this paper, we have dealt with a rule mining problem using genetic algorithms. The experiments conducted show the enhancement of the results compared with (Boujelben et al., 2013) process of about 8% in terms of recall and precision. Some problems related to the relation extraction are resolved, like the segmentation into clauses which ensures the presence of semantic relations between NEs. Furthermore, since our class presents the position of support word that explicit a

[4] The precision P is the number of relevant instances of the system among all the treated instances
[5] The recall R is the fraction of relevant instances that are retrieved.
[6] The combination of the precision and recall for penalizing the very large inequalities between these two measures: (2*P*R) / (P+R).

relation, our process has the capability to extract a several number of relations between NEs.

But, we have not treated the case of relation that is expressed through more than one word which may certainly improve the effectiveness and the understanding of our process.

As perspectives, there are still more points to be improved in our method. First, the addition of rules written by a linguistic expert can improve certainly the efficiency of our system. In fact, such rules can resolve the problem of negative relation as well as relation that occur between more than two NEs. And finally, it would be interesting to evaluate our process on corpus in Latin language in order to test the adaptability of our method.

References

Agrawal, R., Srikant, R., Imielinski, T., and Swami, A. (1993). *Mining Association rules between Sets of items in Large Databases*. ACM., pp.207-216.

Benajiba, Y., Rosso, P. and Benedi, J. M. (2007). *ANERsys: An Arabic Named Entity Recognition system based on Maximum Entropy*. CICLing, Springer-Verlag, Berlin, Heidelberg, pp. 143-153.

Ben Hamadou, A., Piton, O. and Fehri. H. (2010). *Multilingual Extraction of functional relations between Arabic Named Entities using NooJ platform*, hal-00547940, version 1.

Boujelben, I., Jamoussi, S. and Ben Hamadou, A. (2012). *Enhancing Rules based approach for semantic relations extraction between Arabic named entities*. NooJ2012, in INALCO-Paris.

Boujelben, I., Jamoussi, S. and Ben Hamadou, A. (2013). *Enhancing machine learning results for se-mantic relation extraction*. NLDB, University of Salford, Manchester, UK.

Delloye, Y.N. (2010). *Extraction des chemins entre deux entités nommées en vue de l'acquisition des patrons de relations.* IC.

Holland, J.H. (1970). *Robust algorithms for adaptation set in a general formal framework,* Proceedings of the IEEE Symposium on Adaptive Processes - Decision and Control 17.

Hasegawa, T., Sekine, S. and Grishman, R. (2004). *Discovering relations among named entities from large corpora.* ACL Association for Computational Linguistics.

Jourdan, L., Dhaenens, C. and Talbi, E.G. (2002). *ASGARD: un algorithme génétique pour les règles d'association,* Revue ECA, pp. 657-683.

Keskes, I., Benamara, F. and Belguith, L. (2012). *Clause-based Discourse Segmentation of Arabic Texts*. Language Resources and Evaluation LREC, pp. 21-27.

Kramdi, S. E., Haemmerl, O. and Hernandez, N. (2009). *Approche générique pour l'extraction de relations partir de textes.* Ingnierie des Connaissances IC, Tunisia.

Mesfar, S. (2008). *Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en Arabe standard,* University of Franche-Comté.

Quinlan, J. R. (1993). *Programs for Machine Learning.* Morgan Kaufmann Publishers. San Mateo.

Wang, W., Romaric, B., Olivie, F. and Grau, B.(2011). Filtrage des relations pour l'extraction de l'information non supervisé . TALN.

Zhang, Z. (2004). *Weekly supervised relation classification for information extraction*, 13th Conference on Information and Knowledge Management CIKM2004, Washington D.C., USA.