Identifying Non-referential Arabic Pronouns: a Self-training SVM Approach

Saoussen Mathlouthi Bouzid[1] and Chiraz Ben Othmane Zribi[2]

National School of Computer Science, RIADI Lab, University of Manouba, Tunisia
[1]Mathlouthi.saw@gmail.com
[2]Chiraz.zribi@ensi-uma.tn

Abstract. The classification of pronoun as referential or non-referential is necessary for many NLP tasks. However, there are few works interested in this problem in the Arabic language. In this paper, we present a semi-supervised machine learning approach based on a Self-training SVM method for the identification of non-referential pronouns in the Arabic texts. A set of patterns-based and linguistic-based information is used as classification features in our machine learning system. The proposed Self-Training SVM algorithm includes three steps: training, prediction and selection step. It trains SVM classifier on a small set of labeled data, predicts labels of unlabeled data, selects the most accurate and the most informative newly labeled data and adds them to the training dataset. The selection step uses some geometric measures and analyses their relevance on the classification performance. The evaluation of our approach on the training and test data presents good results that can reach up to 96.85%.

Keywords: Non-referential pronoun, anaphora resolution, morpho-syntactic features, semi-supervised learning, self-training, SVM, similarity measures, Arabic.

1 Introduction

For the Pronominal Anaphora Resolution (PAR) task and many other Natural Language Processing (NLP) applications as translation and information retrieval, it is very useful to distinguish the pronouns that have an antecedent from those that have not. This allows to filter the list of non referential pronouns and to avoid in this manner the loss of time in the search for non-existent antecedents. Since Arabic is a morphologically rich language, it is interesting to use the morpho-syntactic information as features for a machine-learning model to classify the pronouns into referential and non-referential. In fact, machine-learning techniques are considered as very effective for NLP problems.

Over the last decades, the semi-supervised learning has appeared as an interesting new direction in machine learning research. It deals with the situation where relatively few labeled training data are available, but a large number of unlabeled data are given. It is directly relevant to a multitude of practical problems where it is relatively expensive to produce labeled data [1]. Indeed, labeled instances are often difficult,

expensive, or time consuming to obtain. While, unlabeled data may be relatively easy to collect, but they are not informative enough. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers [2]. Therefore, semi-supervised learning can be considered as a good compromise between supervised and unsupervised learning.

There are two classes of semi-supervised learning methods: transductive or inductive methods. Transductive learners label only unlabeled examples of the training data and cannot handle unseen data. By cons, inductive learners find a prediction function from the training data, then apply this function to the new test data, so, they can naturally handle unseen data. Different methods of semi-supervised inductive learning have been proposed such as Expectation Maximisation (EM), Co-training, graph-based method, self-training. The self-training algorithm turns around a base learner and selects newly-labeled data at each iteration of the training procedure. So, the strength of the self-training algorithm resides in the confidence prediction measure that allows to extend the small labeled database with more informative newly-labeled data.

Related works on this topic are limited, especially for Arabic language which is characterized by several specifications. Indeed, the morphological and the syntactical ambiguity in Arabic is an important problem for the PAR task. It is due to the agglutination of clitics[1], the diacritical[2] marks and the exceptional cases of gender and number agreement. Also, Arabic Language is considered as a low-resource language. So, the lack of Arabic resources for NLP, such as annotation tools, labeled training corpora and test corpora, make the task more difficult.

The limited works done on the identification of non-referential anaphors has motivated us to propose a novel approach. Our proposed approach is based on a semi-supervised self-training learning method. It uses an SVM classifier and operates on a set of patterns-based and linguistic-based features. Our main goal is to increase labeled database with newly-labeled data, and to improve the performance of the SVM classifier. The selection of the newly-labeled data is based on two steps: a first step allows to choose the most accurate data and a second step enables to distinguish the most informative ones. The evaluation of our approach on the training and test data gave us promising results.

The remaining of this paper is organized as follows. In Section 2, we present a linguistic study to identify the most non-referential pronouns constructions in Arabic texts. Pattern constructions and other morpho-syntactic information are used as features for the proposed machine learning method. Previous approaches are described in details in Section 3. In Section 4, we explain our proposed semi-supervised approach. Finally, in Section 5, we present the datasets, the results of experiments, and a comparison with similar works.

---

[1] Clitics are elements of grammar attached to a root of a word.

[2] Short vowels in Arabic are replaced by symbols called diacritics.

2 Classification features

In Arabic language, there are many interesting cases of pronouns that do not refer to anything. We achieved a linguistic study in Arabic texts to identify the constructions of non-referential pronouns. For the selection of the classification features, one question we had to address was which information from the text we wanted to use. In one hand, the information capturing the left and right context of pronoun must be expected to help in the classification task. In the other hand, the classification system widely depend on the non-referential patterns. Thus, we chose to use linguistic-based and pattern-based information together as classification features in our machine learning system. Each instance of pronoun is represented as a vector of features (attribute-value pairs). The linguistic-based features include grammatical and syntactical features. The pattern-based features test the verification of the non-referential patterns described below.

2.1 Pattern-based features

The most-known models of non-referential constructions can be presented as patterns. We have identified 10 patterns classified into four groups described in Table 1. If a pronoun checks one of these patterns then it is certainly (or most probably) a non-referential pronoun.

Table 1. Non-referential patterns features

	Features
Confirmation patterns	إنّه مِنَ (غير) + defined adjective إنّه مِنَ الطبيعي التعايش مع الآخرين (1) (It is natural to live with others)
	إنّه+ Specific delimiter + مِنْ أنْ + verb إنّه لا مفر منْ أنْ أكرّر بعض الحجج (2) (It is inevitable to repeat some arguments)
	إنّه مَنْ + verb / أنّه مَنْ + verb / لعلّه مَنْ + verb إنّه من يتق الله يجعل له مخرجا (3) (Whoever fears God makes him a way out)
	إنّه + لولا + defined noun إنّه لولا الجار لهلك (4) (Without the neighbor, he would have perished)
Time and climate patterns	إنّه / إنّها + specific climate or atmosphere verb إنّها تمطر كثيراً في هذا الوقت من السّنة (5) (It rains a lot at this time of year)
	إنّها + number (hour/time) + Specific words إنّها الثّالثة فجرا، البيت هادئ جدّا(6) (It is three o'clock in the morning, the house is very quiet)
	إنّه منذ 50 عاما أو أكثر والبلاد تكافح من أجل النّهوض (7) (It's been 50 years or more and the country is struggling to advance)
Proverb patterns	Proverbs and sayings لا تحملها في قلبك (10) (Do not carry it in your heart)
Other patterns	ما + verb+ attached pronoun ما أنتجته هذه الفترة كان أهم النتائج(8) (What produced this period was the most important results)
	هناك + مازال /لا يزال + nom لا يزال هناك عمل كثير (9) (There is still a lot of work)

2.2 Linguistic-based features

Linguistic-based features include grammatical and syntactical features. Grammatical features indicate the grammatical value, the gender and the number of the current pronoun and of the words surrounding it. The syntactical features, mark important syntactical characteristics. Table 2 details this type of features.

Table 2. Non-referential linguistic features

N°	Features	Interpretation
		Grammatical features
1.	Vg, Gr, Nbr,	Grammatical value (Vg), gender (Gr) and number (Nbr) of the current word, the enclitic (Enc) attached to current word, the two previ-ous (Prev) words and the three next (Suiv) words surrounding the pronoun.
2.	VgEnc, GrEnc, NbrEnc,	
		Syntactical features
3.	Pos_Pron	Position of the pronoun in the sentence
4.	Exist_Delim	Existence of a discriminating delimiter that immediately fol-lows the pronoun
5.	Dist_Pr_Delim	Distance from the pronoun to the first delimiter
6.	Agreement	Gender and number agreement between the pronoun and the following verb
7.	Follow_Noun	Existence of a definite noun that immediately follows the pronoun
8.	Follow_Part_Spec	Existence of a specific particle that follows the pronoun
9.	Impersonal_Verb	Existence of an impersonal verb after pronoun
10.	Pron_Demonst	The current pronoun is demonstrative

3 Related work

The problem of identifying non-referential (impersonal or pleonastic) pronouns is the subject of several works especially for English and French and with a lesser degree for Arabic. Recognition of impersonal "it" pronouns has begun since the 1980s with Katzer et al. [3] and Paice and Husk [4]. This latter [4] says: "Clearly, before looking for an antecedent, it is essential to know whether a word really is anaphoric". The proposed approaches can be categorized into rule-based, learning-based or hybrid approaches.

Rule-based approaches are the first types of approaches used in several works especially in the works [4],[5],[6],[7] for English, [8] for French and [9],[10] for Arabic language. The Paice and Husk approach [4] have applied an appropriate set of contextual rules to determine whether the use of the "it" is anaphoric or not. They illustrated the different ways of using the pronoun "it" and the rules that build them. Lappin and Leass [5] have used a set of constructs to test the presence of pleonastic "it". The tests are both syntactic and lexical. For the Arabic language, Elghamri et al. [9] worked with a set of heuristics for pronoun recognition. One of the examples of

pleonastic pronouns is the pronoun "ه" (he / he / her) attached to the particle أن / èn / which disappears following an Arabic-English translation. While Mathlouthi et al. [10] proposed an approach based on linguistic rules. The rules include constructs inspired from the Lappin and Leass work [5] and adapted to the Arabic language.

Learning-based approaches are more recent. The researches have proposed different methods of learning namely the K-Nearest-Neighbor method (KNN), the Support Vector Machine (SVM) method, the maximum entropy method. They have used different classification characteristics concerning syntactic patterns, lexical characteristics, distance characteristics, syntactic information around the context, position characteristics and other classification features. Evans [11] proposed a KNN learning method for the classification of pronoun “it”. He operated 35 classification features relating to syntactic characteristics and surface clues. Bergsma et al. [13] have proposed a distributed approach that extracts the context of the pronoun "it" then looks for models whose words can replace this pronoun in the same context. Those models are extracted from n-gram collection and the classification used the maximum entropy method.

Learning-based approaches use learning techniques to automatically produce non-referential constructs and ignore the manual production of linguistic rules. While, rule-based approaches benefit from linguistic knowledge and translate it into well-defined syntactical rules. However, hybrid approaches can benefit from the advantages of the last two approaches to achieve better results. There are several hybrid approaches such as: [14], [15], [16] for English language, [17], [18] for the Arabic language. So, Müller [14] have used 38 classification features through JRip rule-learner. The features consist on 21 surface syntactic patterns, lexical information about the predicative context of “it”, features that capture the wider context from “it” to certain categories of words and binary features encoding whether the “it” verify some rules. Weissenbacher and Nazarenko [16] have identified impersonal pronouns in English texts by combining linguistic knowledge and surface clues. They use a Bayesian Network method presented as a probabilistic graph model applying rules inspired from the woks of Lappin and Leass [5] and Paice and Husk [4]. Hammami et al. [18] have proposed an approach inspired from the work of Weissenbacher and Nazarenko [16]. They have used a method based on Bayesian networks to identify pleonastic pronouns in Arabic texts. They have proposed a set of general and specific rules to detect the non-referential occurrences of the pronoun "ه" / "hu" ("he").

4 Semi-supervised self-training SVM method

4.1. Self-training process

The self-Training SVM algorithm is based on an SVM classifier. This latter is first trained on a small set of labeled data (the initial training corpus). Next, it is used to predict labels of unlabeled examples. A subset of unlabeled examples, with their predicted tags, is selected to increase the initial labeled training set. Then, the classifier is newly trained on the recent training data and used to classify other unlabeled examples. This process is repeated several times until all unlabeled data are processed or a

maximum number of iterations is reached. The architecture of this process is illustrated by Figure1.

Each iteration includes three steps:

- Training step: the SVM classifier is trained on the labeled data. It maps the original input space into a higher dimensional feature space using a certain kernel function. In the new feature space, the SVM algorithm searches the optimal separating hyperplane with maximal margin in order to minimize an upper bound of the expected risk instead of the empirical risk [20].
- Prediction step: the trained classifier is used to classify the unlabeled data and to predict their labels. Each newly-labeled data has an estimation probability used as a confidence measure.
- Selection step: From the obtained predictions, the system selects only the most accurate and the most informative instances and then adds them to the labeled data. As a first step of selection, it is important to retain only the instances for which the prediction probability of the class is high. But, a very high prediction probability selection does not guarantee that the actual class is correctly predicted. Therefore, we use a second step of selection that keeps the most informative data. In the second step, we apply different similarity measures described in Section 4.4.

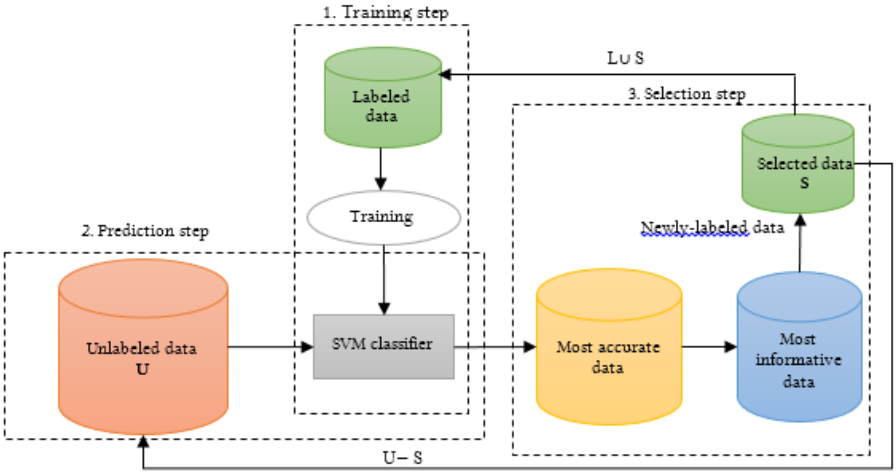

Fig. 1. Semi-supervised self-training process

4.2 Selection step

The most accurate unlabeled data. The self-training process provides, at each iteration, the predicted label of unlabeled data and their corresponding estimated probabilities. To select the high confidence predictions data, we must have a judgment criterion. Indeed, it is interesting to choose the cross validation precision of the classifier as a threshold to judge high confidence predictions. This choice can be explained by the fact that estimated probabilities always depend on the performance of the trained

classifier. Thus, the cross validation precision changes through iterations and it expresses the performance of newly-trained classifier.

The most informative unlabeled data. The estimated probability is used to select high-confidence predictions, which may not always be optimal because of some misclassified examples. Another way to select from the unlabeled examples is to use a second selection step. Once the most accurate newly-labeled data are selected, the second step process to select the most informative one by using geometric measures: Euclidean distance measure or similarity cosine measure. This informative selection step helps to select the most reliable data.

Euclidean distance based measure. The distance-based measure is the Euclidean distance between an unlabeled data point and all positively (+) or all negatively (-) labeled data. Considering X $\boldsymbol{\epsilon}$ U is the feature vector of unlabeled data point, the size of feature vector is the number of classification features n. C1= $\{c1_i\}_{i=1..n}$ is the centroid of negatively labeled data. C2=$\{c2_i\}_{i=1..n}$ is the centroid of positively labeled data. The Euclidean distance d1 (respectively d2) between X and the centroid C1 (respectively C2) are defined by the formulas (1) and presented in Figure 2.

$$d=\sqrt{\sum_{i=1}^{n}(c_i - x_i)^2} \quad (1)$$

The centroid vector of each class is defined by using SimpleKMeans weka method. This method is a clustering approach based on centroids computation. We use the WEKA implementation of the clustering KMeans in Java to define the class centroid vectors. The difference between calculated distances (d1-d2) gives more information about the nearest class to the point data. If the subtraction of the distances exceeds a threshold (determined empirically) then the data point is chosen as a good informative data.

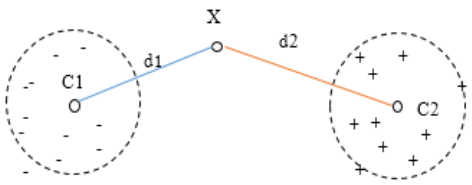

Fig. 2. Euclidean distance from unlabeled data X to negative data centroid and positive data centroid

Similarity cosine based measure. The similarity cosine based measure introduces the cosine value of data point X in the predicted class. The computation of cosine value uses the feature vector X of unlabeled data point and the centroids (C1, C2) of each class. The point data X can be discarded or chosen according to the value of the similarity cosine as shown in Figure 3.

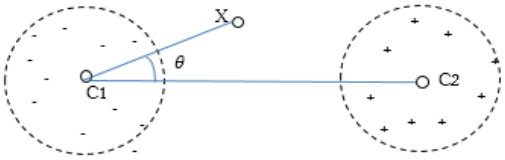

Fig. 3. Similarity cosine of unlabeled data X and negative and positive data centroid

The formula (2) describe the similarity measure between two vectors; the first vector is formed by the two class centroid points C1 and C2, the second vector is formed by the data point X and the centroid point of predicted class. Thus we must privilege the points forming large radius (small cosine value), they must be the farthest from the positively labeled class and subsequently closer to the negatively labeled class, or vice-versa. So, the data point, whose cosine value is less than the given threshold, is selected and added to labeled data then used to re-train again the SVM classifier.

$$\cos\theta = \frac{\overrightarrow{c_1c_2}\,\overrightarrow{c_1x}}{\left\|\overrightarrow{c_1c_2}\right\|\left\|\overrightarrow{c_1x}\right\|} \tag{2}$$

4.3 Self-training algorithm

The self-training algorithm uses a small set of labeled data L= {Xi, Yi}; for each point Xi the label Yi ϵ{-1,1} is known (since "1" is the label of the referential class and "-1" is the label of the non-referential class). As well, the algorithm uses a large number of unlabeled data U= {Xi} for which the labels are unknown. As already said, the self-training algorithm uses SVM to classify the unlabeled data and to predict their labels. Next, a subset of unlabeled examples are selected after two filtering steps. The first filtering step determines the most accurate data. It is based on high estimation probabilities of the predicted labels which must exceed a well-chosen threshold. The second filtering step is based on a geometric measure that allows to choose the most informative data. The data set S, selected after filtering, are added to the labeled data set L = L∪S. The classifier is then re-trained on the new set of labeled examples. The process is repeated until it reaches a stopping condition. The resolution process is implemented according to the Algorithm1.

```
Algorithm 1: pseudo-code of self-training SVM algorithm
Input: L: labeled data; U: unlabeled data; S: selected
data; T1: 1st selection threshold; T2: 2nd selection
threshold; NbrIter: maximum number of iteration
Output: New classification model trained on larger la-
beled data
Initialize: S=∅; t←0; m=0
//t: counter on number iteration; m: geometric measure of
data
While (U != ∅ ) and (t< NbrIter)
```

```
svm.buildClassifier(L)  // train SVM classifier on la-
beled data
T1= crossValidateModel.precision  // the 1st selection
threshold is the cross
// validation precision of the training classifier
For each Xi ϵ U do
Yi= svm.classifyInstance(Xi) // predicted label of Xi
Pi= svm.distributionForInstance (Xi)  // estimate proba-
bility of predicted label
If (Pi > T1) then
m=Geo-Measure(Xi)  // geometric measure calculated with
Euclidean
      // distance or cosine similarity
If (m > T2) then
S=S ∪ {Xi}
End if
End if
L = L∪S                        // Update L and U
U = U−S
End for
Re-trained SVM classifier on new labeled data L
t ← t+1
End while
```

The algorithm 1 processes training, prediction and selection step. Selection step handles instance by instance and chooses only instances that check both conditions and verify the two filter thresholds. For each iteration, the SVM classifier is re-trained on newly-labeled data. The finally trained classifier can be used for test data.

5 Experimental results

To measure the efficiency of the proposed approach, we achieved different experiments. Firstly, we evaluated the improvement of our method through the self-training iterations. Secondly, we compared the results of the self-training approach using either distance or cosine similarity measures. Likewise, we evaluated the proposed approach on training and test data.

5.1 Datasets

We used a corpus of literary texts extracted from children's stories and a Tunisian basic education textbook. The experimental data set includes the training data and the test data. The training data consists of a small set of labeled data and a big set of unlabeled data. It is used to train the SVM learning classifier. The labeled data are the initial data set for self-training algorithm. The test data are used to finally evaluate the

performance of the proposed approach. The size of the train and the test data is given by Table 3.

The big part of the data is used to train classifier. The rest is kept as test data. For training data, 95.5% of data instances are unlabeled while 4.5% are labeled. Usually, the number of referential pronouns is much larger than the number of non-referential pronouns. For labeled data, we tried to use a data set balanced in number of referential and non-referential pronouns; this to provide a better classification of unlabeled instances. For unlabeled data, the size of data is quite large, and it is difficult to provide a balanced number of data for the two pronoun classes. Then, we proceed to apply the Weka SMOTE3 filter to create new instances of non-referential data.

Table 3. Datasets size in terms of word and pronoun number

Data size	Words		Pronouns			
Training data	10877	1525	Labeled	68	Non-Referential	30
					Referential	38
			Unlabeled	1457	Non-Referential	31
					Referential	1426
Test data	436	67			Non-Referential	31
					Referential	36

5.2 Experiments and results

Our evaluation gives result about learning precision and test precision. The learning precision is the Cross Validation precision of the classifier at each iteration using training data. The test precision is the final self-training SVM precision on test data according to the chosen informative threshold. We studied the impact of the informative selection step and we conducted comparison of two informative methods: distance based method and cosine similarity method. We tried to identify the best informative method and the corresponding threshold that achieve the good performance.

Learning precision improvement.

. For evaluating the performance of the classifier at each iteration of the self-training algorithm, we used the 10-fold cross-validation method. This method provides an average accuracy of the SVM classifier, so it detects the classifier improvement. However, it is clear that the selection of the most accurate and most informative data has a direct impact on the performance at each iteration. Figure 4 and 5 show the cross-validation precision of the classifier through the self-training process iterations using distance-based and cosine similarity measure.

As explained before, at each iteration of the self-training algorithm a set of informative newly-labeled examples is selected for the next iterations. Therefore, we tested different thresholds for the distance-based measure in order to be able to select

[3] The filter resamples a dataset by applying the Synthetic Minority Oversampling TEchnique (SMOTE). The amount of SMOTE and the number of nearest neighbors may be specified as needed in order to balance the two-class instances size

the most informative data subset. We noticed that the best thresholds are between 10 and 20. The curves of Figure 4 show the SVM learner improvement through the self-training process for the distance based measure.

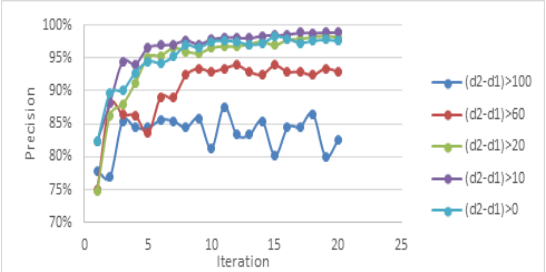

Fig. 4. : Cross validation precision of SVM Classifier at each self-training iteration with distance-based measure

Likewise, we tested different threshold for cosine similarity value. The best thresholds were between 0.1 and 0.3. The curves in Figure 5 show the SVM learner improvement through self-training process for cosine similarity measure.

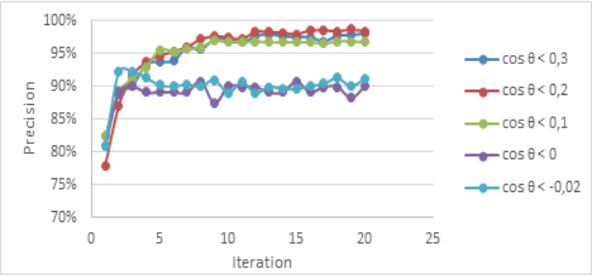

Fig. 5. : Cross validation precision of SVM Classifier at each self-training iteration with cosine similarity measure

Self-training SVM precision

. The performance of the proposed self-training SVM approach depends on several parameters. Firstly, the classification feature number affects the results. The available features that indicate the properties of the context around the pronoun are quite large. Accordingly, we selected the most interesting features. Let us note that it is difficult to predict perfectly the best features because it always depends on the context of the treated text. Secondly, the stopping conditions and the iteration number of the self-training process can also impact on the whole performance. To reach the best-trained classifier, we achieved experiments in order to find suitable iteration number for performance improvement. Finally, we noticed that the chosen informative selection method and the correspondent thresholds strongly influence the precision results. The results of the experiments shown in Table 4, 5 and 6 confirm this. We performed sev-

eral statistical tests to show the effectiveness of the proposed methods. The following evaluations were perform on test data.

Table 4 shows the performance of the semi-supervised self-training SVM approach without informative selection step. The results indicate the precision of Non-Referential class, the precision of Referential class and the average precision of the two last values. We noticed that the precisions are quite performing but not good enough.

Table 4. Approach precision without informative side

Precision	Precision of Non-Referential class	Precision of Referential class
83.75%	87.5%	80%

Table 5 gives results of the self-training SVM with informative selection step based on distance measure. We presented the average precision of the approach and the precision of each class. The more the difference between distances is greater, the more the chosen instances are informative and can improve results. Depending on obtained results, the selection of the most informative data, that having a distance variation greater than 60, increase precision. Consequently, the choice of the threshold has a direct impact on the performance.

Table 5. Approach precision according to the distance measure threshold

(d1-d2) > Threshold	Precision	Precision of Non-Referential class	Precision of Referential class
d1 - d2 > 10	84,29%	90%	78,6%
d1 - d2 > 20	88.23%	93.5%	82,9%
d1 - d2 > 40	84.07%	92.6%	75.6%
d1 - d2 > 60	90%	96.7%	83.3%
d1 - d2 > 90	90%	96.7%	83.3%

Similarly, Table 6 gives results of the self-training SVM on test data when the informative selection step is based on cosine similarity measures. The precisions strongly depend on the chosen threshold. Noting that $cos\ \theta$ belongs to [-1, 1], the low cosine correspond to large radius. So, the more cosine value is lower the more the selected instances are informative. Accordingly, the experiments showed that the best way is to choose an average threshold value approximated to 0.1.

Table 6. Approach precision according to cosine similarity threshold

$cos\ \theta$ < Threshold	Precision	Precision of Non-Referential class	Precision of Referential class
$\boldsymbol{cos\ \theta}$ < 0.3	83.2%	89.7%	76.7%
$\boldsymbol{cos\ \theta}$ < 0.2	84.29%	90%	78.6%
$\boldsymbol{cos\ \theta}$ < 0.1	90%	80.6%	97.2%
$\boldsymbol{cos\ \theta}$ < 0	83.75%	87.5%	80%
$\boldsymbol{cos\ \theta}$ < -0.1	86.1%	86.1%	86.1%
$\boldsymbol{cos\ \theta}$ < -0.02	85.41%	90.3%	80.5%

The experiment results showed that the good choice of the selection threshold improves the SVM classifier learning and produces better classification model. So, despite the lack of labeled data, our results showed that such an approach can give good results.

5.3 Discussion and comparison results

As mentioned before, several works have treated the identification of non-referential pronoun. Most of the works deal with the English language but very few researchers were interested in the Arabic language. To have a meaningful comparison, we compared our approach to similar work for Arabic language in Table 6.

Table 7. Comparison with similar approach

Work	Type of approach	Precision	Corpus
Elghamri et al. [9]	Rule-based approach	-	-
Mathlouthi et al. [10]	Rule-based approach	-	-
Abdul-Mageed [17]	Hybrid approach using supervised learning based on the MBL algorithm	97%	Sub-segment of the Penn Arabic Treebank containing 721 pronouns in which 100 are non-referential
Hammami [18]	Hybrid approach based on the work of Weissenbacher and Nazarenko [16]	88.58%	Corpora of different domain containing 4323 pronouns in which 1747 are non-referential
Our approach	Hybrid approach using semi-supervised learning based on Self-training SVM	96.7%	Corpus of literary texts containing 1592 pronouns in which 92 are non-referential

The size and the type of the corpus have a considerable influence on the evaluation results. The efficiency of approach depend on the complexity of the treated text and the size of the labeled data. The approach of Abdul-Mageed [17] gave a good result on a hundred of labeled data. The supervised method is effective but it cannot handle unlabeled data. The approach of Hammami [18] based on the work of Weissenbacher and Nazarenko [16] used Bayesian network method. The supervised method is less performing than that of Abdul-Mageed [17] but it used a larger corpus. For our approach, we proposed a self-training method using few labeled and much unlabeled data. Our semi-supervised learning approach gave a result close to that of Abdul-Mageed [17], as well it allows to extend the set of labeled data. Moreover, our approach outperforms the results of Hammami et al. [18], whereas their corpus is wider. Despite the lack of labeled data, we found satisfying results. Our proposed approach offers an automatic increase of the labeled database that can be used later to solve new test data.

6 Conclusion

In this paper, we presented a semi-supervised self-training SVM approach for the classification of referential and non-referential pronouns in Arabic texts using linguistic features and patterns-based features. We was able to increase the set of labeled training data and to improve classification and we overcame by this way the big problem of the lack of labeled data. In addition, we obtained good performances on training and test data, the precision reached up to 96.7%.

However, the main weakness of our approach is that it remains difficult to provide a balanced number of referential and non-referential pronouns, because usually the number of referential pronouns is much larger than the number of non-referential, and this may impact on the classification task. In the future, we aim to increase the number of non-referential pronouns as much as possible. We can also consider not only the distance or cosine measure methods but also other geometric-based methods. As well, we plan to build a larger corpus and enrich it with other types of texts.

References

1. Chapelle, O., Schölkopf, B. and Zien, A.: "Semi-Supervised Learning", Massachusetts Institute of Technology, The MIT Press Cambridge, Massachusetts London, England. (2006)
2. Zhu, X., "Semi-Supervised Learning Literature Survey",Computer Sciences TR 1530, University of Wisconsin – Madison, Last modified on July 19, (2008).
3. Katzer, J.. Bonzi, S. & Liddy, E.. Impact of Anaphora in Information Retrieval. Final report to National Science Foundation, School of Information Studies, Syracuse University. Syracuse. New York. (1986)
4. C. D. Paice and G. D. Husk, Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun "it", Department of Computing. University of Lancaster, U.K. Computer Speech and Language 2, 109. 132. (I987).
5. S. Lappin, H. J. Leass, "An Algorithm for Pronominal Anaphora Resolution". London, Computational Linguistics, 20(4), 535-561, (1994).
6. M. Denber, "Automatic Resolution of Anaphora in English", Eastman Kodak Co. Imaging Science Division June 30, (1998)
7. F.-R. Chaumartin, "Résolution d'anaphores dans une encyclopédie en langue anglaise : conception, implémentation et évaluation des performances", Laboratoire LATTICE – Université Paris 7, (2007).
8. L. Danlos, «ILIMP : Outil pour repérer les occurrences du pronom impersonnel il », LATTICE, Université Paris 7, Institut Universitaire de France, TALN 2005, Dourdan, 6-10 juin (2005).
9. K. Elghamry, R. Al-Sabbagh and N. El-Zeiny, "Arabic Anaphora Resolution Using Web as Corpus", Proceedings of the seventh conference on language engineering, pp. 1-18. Cairo, Egypt, (2007).
10. S. Mathlouthi, F. Ben Fraj Trabelsi, C. Ben Othmane Zribi, "A Novel Approach Based on Reinforcement Learning for Anaphora Resolution", 28th IBIMA Conference, November (2016).
11. Richard Evans. "Applying machine learning toward an automatic classification of it". Literary and linguistic computing, 16 N°1: 45--57, 30, 31, 32, 111. (2001)

12. Litrán, J. C., Satou, K., and Torisawa, K., “Improving the identification of non-anaphoric it using support vector machines”. (NLPBA/BioNLP), pages 61--64, Geneva, Switzerland, August 28th and 29th. COLING. 31, 33, 115. (2004).
13. Bergsma, S., Lin, D., and Goebel, R., “Distributional identification of non-referential pronouns”. In Proceedings of Association for Computational Linguistics ACL-08 : HLT, pages 10--18, Columbus, Ohio, USA, 31, 109. (2008).
14. Müller, C., “Fully Automatic Resolution of It, This and That in Unrestricted Multi-Party Dialog”, von Mark-, Philosophische Dissertation angenommen von der Neuphilologischen Fakultät der Universität Tübingen am 12. Juin Tübingen. (2008).
15. Boyd, A., Gegg-Harrison, W., and Byron, D. “Identifying non-referential it: a machine learning approach incorporating linguistically motivated patterns”. In ACL Workshop on Feature Engineering for Machine Learning in NLP, pages 40--47, 32, 109. (2005)
16. Weissenbacher, D., and Nazarenko, A.: “A bayesian classifier for the recognition of the impersonal occurrences of the it pronoun”. In Proceedings of DAARC’07, 29, 32, 43, 72, 120. (2007).
17. Abdul-Mageed, M.: “Automatic detection of Arabic non-anaphoric pronouns for improving anaphora resolution”. ACM Transactions on Asian Language Information Processing (TALIP), 10 N°1 :535--561, 33, 108. (2011).
18. S. Hammami, “La résolution automatique des anaphores pronominales pour la langue arabe”, thèse de doctorat, Université de Sfax, Faculté des Sciences Economiques et de Gestion, Sfax, Tunisie, (2016).
19. C. Hechiri. The pronoun and its role in the sentence. PhD thesis. Faculty of Letters and Humanities of Sfax, Sfax, Tunisia. (1998).
20. I. Triguero, Sa. García, F. Herrera: “Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study”, Springer-Verlag London, DOI 10.1007/s10115-013-0706-y. (2013).