An Optimized Method for Arabic Cross-Document Named Entity Normalization

Khaled S. Refaat and Amgad Madkour

IBM Technology Development Center
Pyramids Heights, Office Park
Egypt
{ksaeed,amadkour}@eg.ibm.com

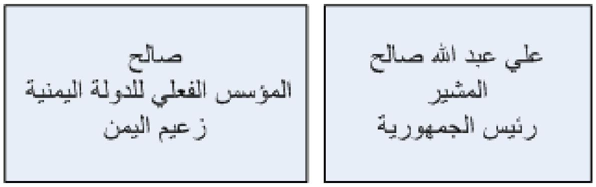
Abstract

This paper presents a technique to perform Arabic cross-document named entity normalization. The proposed method offers significant time improvement over conventional *nxn* comparisons performed between named entities. It relies on a novel efficient algorithm that avoids normalizing the new entities against all existing entities. Only a single candidate from the normalized entities is chosen to be checked against each new entity. This allows using extensive normalization checking only with the entity that is most likely to be normalized. Our results show that we obtain comparative results in nearly half the time required by conventional named entity normalization methods. We have also tuned a SVM model that decides whether two entities should be merged or not. This SVM model outperforms the related work in accuracy by 9%.

Introduction

News is a rich source of information that can aid in discovering new types of relations between entities such as people. One of the main challenges is to determine if two names are used to denote the same person. The fact that a person's name can be expressed in variable forms makes it hard to normalize entities. Some people are referred to by their first name and others by their last name. Sometimes they are referred to by a commonly known name, especially in Arab countries, such as the name "Abo Mazen" or "Abo Alaa" which means "Father of Mazen" or "Father of Alaa". Most contributions have focused on performing comparison between a name and another candidate name in order to determine their similarity. Most methods depended on syntactic structures which could be measured by edit distance for instance. Other methods include comparison based on the semantic features that describe the name.

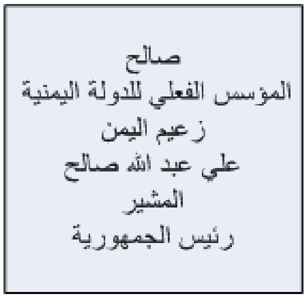
Each news document consists of a number of entities. Each entity represents a single person and is composed of all his name and nominal mentions that are present in the document. Named entity normalizations across documents is to decide which entities, of different documents, are representing the same person and therefore should be normalized as one normalized entity. The aim is to group their name and nominal mentions in one entity. This cross-document normalization requires checking each pair of entities in order to judge whether they should be normalized as one entity or kept separate. Figures 1 and 2 show an example of two entities that belong to the same character (Abdullah Saleh, the current president of Yemen), and their form after being merged respectively.



صالح
المؤسس الفعلي للدولة اليمنية
زعيم اليمن

علي عبد الله صالح
المشير
رئيس الجمهورية

Figure 1: Two entities belonging to the same person (Abdullah Saleh)



صالح
المؤسس الفعلي للدولة اليمنية
زعيم اليمن
علي عبد الله صالح
المشير
رئيس الجمهورية

Figure 2: The merged entity of Abdullah Saleh

A number of challenges exist in named entity normalization. First, an exhaustive comparison is needed in order to perform the normalization. Second, normalization should be done across documents. This generates a major problem given the immense amount of documents available where each contains a significant amount of entities and mentions. The extensive checking is computationally very expensive especially when we are working with a massive number of entities. Moreover, when this huge number of entities needs to be normalized daily, an efficient normalization method becomes indispensable.

This work proposes an optimized methodology for Arabic named entity normalization. We utilize the name entity normalization methodology proposed by (Magdy et al, 2007). Their system utilized a Support Vector Machine (SVM) to determine if two entities should be normalized.

We propose a way to break the exhaustive *nxn* entities comparisons while obtaining acceptable normalization results. It is based on performing an efficient check between the new coming entity and the already existing ones. This efficient check is a deterministic algorithm that chooses the best candidate from the already normalized entities. The chosen entity is then checked extensively against the new unseen entity using SVM. We have developed a well trained support vector machine for confirming if an entity should join a candidate entity or not. The proposed method allows breaking the time required for normalization in half. The reason of this is that the efficient algorithm used initially allows us to avoid using SVM to check the new entity against all existing entities.

The paper is organized as follows. We first present the related work. Secondly, we explain the idea of SVM in short. We then propose the efficient normalization technique. Finally, the experimental results are illustrated with a discussion. The paper ends with a conclusion.

Related Work

(Cohen et. al, 2005) proposed an unsupervised gene/protein named entity normalization system using automatically extracted dictionaries. They have built a dictionary based gene and protein Named Entity Recognition (NER), and normalization system that requires no supervision and no human intervention in order to create the dictionaries from online genomics resources. They have tested their system on the Genia corpus and the BioCreative Task 1B mouse and yeast corpora and achieved a level of performance comparable to state-of-the-art systems that require supervised learning and manual dictionary creation which is a very costly process. In fact, their technique should also work for organisms following similar naming conventions as mouse, such as human. Further evaluation and improvement of gene/protein NER and normalization systems is somewhat hampered by the lack of larger test collections and collections for additional organisms, such as human. (Jijkoun et al, 2008) proposed five improvements to the baseline NEN algorithm, to arrive at a language independent NEN system that achieves overall accuracy scores of 90% on the English data set and 89% on the Dutch data set. They showed that each of the improvements contributes to the overall score of their improved NEN algorithm, and concluded with an error analysis on both Dutch and English language UGC. Such Modifications are interesting because the NEN system is computationally efficient and runs with very modest computational requirements. (Magdy et al, 2007) proposed a cross-document normalization system that performs extensive *nxn* SVM classification in order to normalize a batch of n entities. This is a computationally expensive procedure. (Wang et al, 2007) evaluates exact, rule-based and various string-similarity-based matching techniques. Their evaluation shows that a rule-based matcher works better on the gold-standard data.

(Lie et al, 2006) provided quantitative assessment of the complexity of BNET on protein entities through BioThesaurus, a thesaurus of gene/protein names for UniProt knowledgebase (UniProtKB) entries that were acquired using online resources.

They evaluated the complexity through several perspectives: ambiguity (i.e., the number of genes/proteins represented by one name), synonymy (i.e., the number of names associated with the same gene/ protein), and coverage (i.e., the percentage of gene/protein names in text included in the thesaurus). They also normalized names in BioThesaurus and measures were obtained twice, once before normalization and once after.

Their study indicated that names for genes/proteins are highly ambiguous and there are usually multiple names for the same gene or protein. It also demonstrated that most gene/protein names appearing in text can be found in BioThesaurus.

Support Vector Machine

The main idea of support vector machine is to a map a given data set into a higher dimensional feature space. The data patterns become linearly separable after such transformation. Accordingly, learning the decision boundary becomes easier. Figures 3 and 4 shows a data set before and after such mapping respectively.

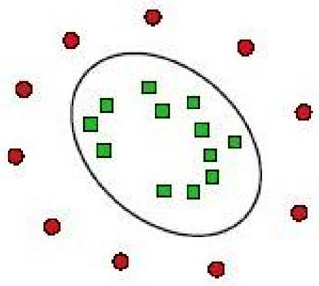


Figure 3: Dataset before mapping

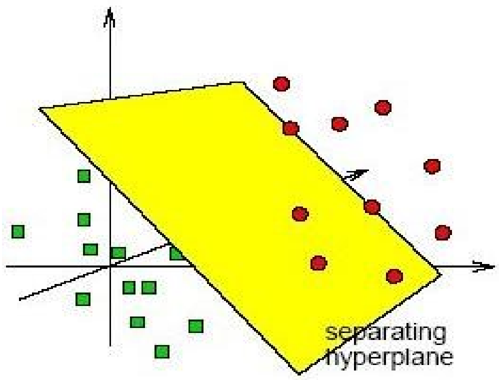


Figure 4: Dataset after mapping

This mapping to a higher dimensional data set is done using a kernel function. There are many types of kernels such as the rbf kernel. The tuning of the parameters of the kernel plays a significant role in increasing the accuracy of the SVM model. In our work we used the same features proposed by (Magdy et al, 2007) for training the SVM that is used to decide whether to merge the two input entities or not. We have only changed the kernel to rbf instead of linear kernel. We used SVM Light in all our simulations.

Optimized Normalization

Our proposed system passes into two phases. In the first phase, we do not have already existing normalized entities. Therefore, we use a support vector machine to judge whether each two entities represent the same person or not. So our system in only its first day is similar to that of (Magdy et. al, 2007).The proposed SVM was trained by a domain expert through manually created training data. The annotator decides whether two entities should be normalized or not. This manually created training set was used to train the SVM. After the first phase, the numbers of entities are decreased in size. However, most entities become larger as they were combined with other entities.

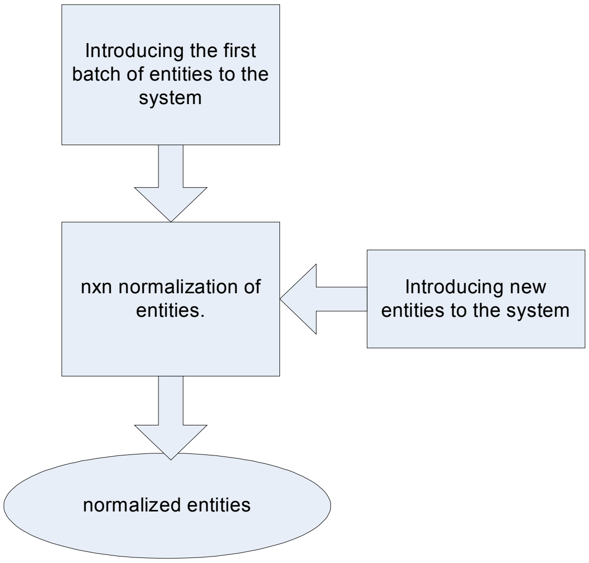

Introducing the first batch of entities to the system
nxn normalization of entities.
Introducing new entities to the system
normalized entities

Figure 5: Magdy *et al* system block diagram.

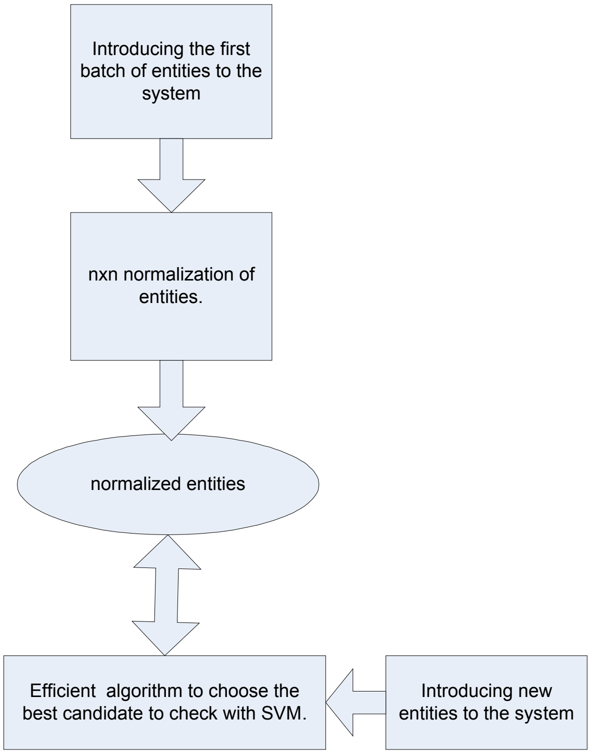

Introducing the first batch of entities to the system
nxn normalization of entities.
normalized entities
Efficient algorithm to choose the best candidate to check with SVM.
Introducing new entities to the system

Figure 6: System block diagram.

In the second phase, a new document introduces new entities to the system. Those new entities are required to be normalized against the already normalized entities. We determine the most frequent name mention of each previously normalized entity to be stored in separate field associated to it. In order to enhance the efficiency of our model, we avoid checking the new entity against all normalized entities using SVM. This is due to the cost associated with such process. Instead, we first perform an initial checking of the new entity against all normalized entities in order to choose the best candidate from among the normalized entities. The initial checking is done with each normalized entity by computing the edit distance of the normalized entity most frequent name mention and all the entity name mentions. If an edit distance is smaller than two we increase the matching score by one. The normalized entity returning the highest matching score is chosen to be checked using the SVM. In other words, the SVM is only used with the normalized entity returning the highest matching score which is more likely to be matched with the new entity using the SVM. Figures 5 and 6 shows a system block diagram of the original (Magdy et al, 2007) and our systems respectively.

We divided the data set into a training set and test set. A SVM model using the RBF kernel with the Gamma parameter set to 2.0 was used to classify the test patterns.

Experimental Results

Our results were compared with the one reported by (Magdy et. al, 2007). We created the same experimental settings proposed by them in terms of SVM features and normalization methodology. Table (1) and (2) illustrate the results.

Model	SVM (1)	SVM (2)
Accuracy	90%	99%

Table 1: Test set accuracy results for the SVM model

Phase/App.	Normal	MF1+MF2	MF1+ED2	MF1+ED3
Phase1Time	42.53	43.9	43.9	43.9
Phase2 Time	71.7	61.4	37.1	38
Phase2 Acc.	99.7%	99.8%	99.8%	99.8%

Table 2: Time and accuracy comparison

Table (1) illustrates the results obtained on an unseen test set between the SVM proposed by (Magdy et al., 2007) denoted by *SVM (1)* and our proposed SVM training denoted by *SVM (2)*. The difference in the results is mainly due to the new parameters we have used in our work. We have used the rbf kernel with the gamma parameter set to 2.0.

Table (2) illustrates the results obtained through the whole normalization cycle. The table also demonstrates how the proposed approach ranks in comparison to the one proposed by Magdi *et al*. All times are given in seconds.

The Normal approach requires a conventional *nxn* comparison which is performed by Magdi *et al*. The Most Frequent (MF) named mention was taken into account with different settings. In the third column, we take the two highest most frequent mentions (MF1+MF2) to be checked against the name mentions of the new entity. Both MF1 and MF2 increase the score by one if they were matching any of the mentions of the new entity. The fourth column presents the results obtained with the most frequent mention and edit distance of 2 (ED2) between the two mentions. Same applies for the fifth column where we compute the most frequent mention and edit distance of 3 (ED3). For all our setups, our results approached 100% in case of Phase 2 while decreasing the computational time significantly.

Discussion

The main reason that the classical method was inefficient is that when the entities normalize with others, they combine to form larger entities. Those bigger entities consist of a larger number of mentions. Therefore, the process of feature extraction for SVM becomes computationally inefficient due to the significant increase in the entity size. It is then better to avoid such costly SVM judgment. Our method uses an efficient algorithm for checking the new entity against the already normalized entities. After that, the algorithm returns a single candidate from the existing entities to be checked using SVM. This new method uses SVM only one time. Whereas the classical method uses SVM with each already existing entity. This causes the system to become sluggish by time since the entities continue to grow in size. The experimental results section shows the significant decrease in time using our proposed technique.

Conclusion

We have proposed a novel approach for efficient named entity normalization across documents. Our technique utilizes an efficient algorithm to avoid exhaustive checks using SVM. The proposed approach decreases the computational time into half in case of using the most frequent mention only. Also the New proposed SVM setting outperforms the old one by 9 %. The accuracy is high in case of seen entities, otherwise the unseen new entities go erroneously to false groups, but the SVM denies the merge.

Bibliographical References

Magdy W., Darwish K, Emam O., Hassan H (2007). Arabic Cross-Document Person Name Normalization. In Proceedings of the workshop on Computational Approaches to Semitic Languages, ACL.

Cohen A. (2005) Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics (pp. 17-24).

Jijkoun V, Khalid M, Marx M, Rijke M (2008) Named Entity Normalization in user generated content. In Proceedings of the second workshop on Analytics for noisy unstructured text data (pp. 23-30).

Wang X. and Matthews M. (2007) Comparing Usability of Matching Techniques for Normalizing Biomedical Named Entities. In Proceedings of the Pacific Symposium on Biocomputing.

Liu H., Hu Z, Torii M., Wu C., Friedman C. Quantitative (2006). Assessment of Dictionary-based Protein Named Entity Tagging. Journal of the American Medical Informatics Association.