Combining Minimally-supervised Methods for Arabic Named Entity Recognition

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio
School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
{mjaltha, udo, poesio}@essex.ac.uk

Abstract

Supervised methods can achieve high performance on NLP tasks, such as Named Entity Recognition (NER), but new annotations are required for every new domain and/or genre change. This has motivated research in minimally supervised methods such as semi-supervised learning and distant learning, but neither technique has yet achieved performance levels comparable to those of supervised methods. Semi-supervised methods tend to have very high precision but comparatively low recall, whereas distant learning tends to achieve higher recall but lower precision. This complementarity suggests that better results may be obtained by combining the two types of minimally supervised methods. In this paper we present a novel approach to Arabic NER using a combination of semi-supervised and distant learning techniques. We trained a semi-supervised NER classifier and another one using distant learning techniques, and then combined them using a variety of classifier combination schemes, including the Bayesian Classifier Combination (BCC) procedure recently proposed for sentiment analysis. According to our results, the BCC model leads to an increase in performance of 8 percentage points over the best base classifiers.

1 Introduction

Supervised learning techniques are very effective and widely used to solve many NLP problems, including NER (Sekine and others, 1998; Benajiba et al., 2007a; Darwish, 2013). The main disadvantage of supervised techniques, however, is the need for a large annotated corpus. Although a considerable amount of annotated data is available for many languages, including Arabic (Zaghouani, 2014), changing the domain or expanding the set of classes always requires domain-specific experts and new annotated data, both of which demand time and effort. Therefore, much of the current research on NER focuses on approaches that require minimal human intervention to export the named entity (NE) classifiers to new domains and to expand NE classes (Nadeau, 2007; Nothman et al., 2013).

Semi-supervised (Abney, 2010) and distant learning approaches (Mintz et al., 2009; Nothman et al., 2013) are alternatives to supervised methods that do not require manually annotated data. These approaches have proved to be effective and easily adaptable to new NE types. However, the performance of such methods tends to be lower than that achieved with supervised methods (Althobaiti et al., 2013; Nadeau, 2007; Nothman et al., 2013).

We propose combining these two minimally supervised methods in order to exploit their respective strengths and thereby obtain better results. Semi-supervised learning tends to be more precise than distant learning, which in turn leads to higher recall than semi-supervised learning. In this work, we use various classifier combination schemes to combine the minimal supervision methods. Most previous studies have examined classifier combination schemes to combine multiple supervised-learning systems (Florian et al., 2003; Saha and Ekbal, 2013), but this research is the first to combine minimal supervision approaches. In addition,

Transactions of the Association for Computational Linguistics, vol. 3, pp. 243–255, 2015. Action Editor: Ryan McDonald.
Submission batch: 1/2015; Revision batch 4/2015; Published 5/2015.

we report our results from testing the recently proposed Independent Bayesian Classifier Combination (IBCC) scheme (Kim and Ghahramani, 2012; Levenberg et al., 2014) and comparing it with traditional voting methods for ensemble combination.

2 Background

2.1 Arabic NER

A lot of research has been devoted to Arabic NER over the past ten years. Much of the initial work employed hand-written rule-based techniques (Mesfar, 2007; Shaalan and Raza, 2009; Elsebai et al., 2009). More recent approaches to Arabic NER are based on supervised learning techniques. The most common supervised learning techniques investigated for Arabic NER are Maximum Entropy (ME) (Benajiba et al., 2007b), Support Vector Machines (SVMs) (Benajiba et al., 2008), and Conditional Random Fields (CRFs) (Benajiba and Rosso, 2008; Abdul-Hamid and Darwish, 2010).

Darwish (2013) presented cross-lingual features for NER that make use of the linguistic properties and knowledge bases of another language. In his study, English capitalisation features and an English knowledge base (DBpedia) were exploited as discriminative features for Arabic NER. A large Machine Translation (MT) phrase table and Wikipedia cross-lingual links were used for translation between Arabic and English. The results showed an overall F-score of 84.3% with an improvement of 5.5% over a strong baseline system on a standard dataset (the ANERcorp set collected by Benajiba et al. (2007a)).

Abdallah et al. (2012) proposed a hybrid NER system for Arabic that integrates a rule-based system with a decision tree classifier. Their integrated approach increased the F-score by between 8% and 14% when compared to the original rule based system and the pure machine learning technique. Oudah and Shaalan (2012) also developed hybrid Arabic NER systems that integrate a rule-based approach with three different supervised techniques: decision trees, SVMs, and logistic regression. Their best hybrid system outperforms state-of-the-art Arabic NER systems (Benajiba and Rosso, 2008; Abdallah et al., 2012) on standard test sets.

2.2 Minimal Supervision and NER

Much current research seeks adequate alternatives to expensive corpus annotation that address the limitations of supervised learning methods: the need for substantial human intervention and the limited number of NE classes that can be handled by the system. Semi-supervised techniques and distant learning are examples of methods that require minimal supervision.

Semi-supervised learning (SSL) (Abney, 2010) has been used for various NLP tasks, including NER (Nadeau, 2007). 'Bootstrapping' is the most common semi-supervised technique. Bootstrapping involves a small degree of supervision, such as a set of seeds, to initiate the learning process (Nadeau and Sekine, 2007). An early study that introduced mutual bootstrapping and proved highly influential is (Riloff and Jones, 1999). They presented an algorithm that begins with a set of seed examples of a particular entity type. Then, all contexts found around these seeds in a large corpus are compiled, ranked, and used to find new examples. Pasca et al. (2006) used the same bootstrapping technique as Riloff and Jones (1999), but applied the technique to very large corpora and managed to generate one million facts with a precision rate of about 88%. AbdelRahman et al. (2010) proposed to integrate bootstrapping semi-supervised pattern recognition and a Conditional Random Fields (CRFs) classifier. They used semi-supervised pattern recognition in order to generate patterns that were then used as features in the CRFs classifier.

Distant learning (DL) is another popular paradigm that avoids the high cost of supervision. It depends on the use of external knowledge (e.g., encyclopedias such as Wikipedia, unlabelled large corpora, or external semantic repositories) to increase the performance of the classifier, or to automatically create new resources for use in the learning process (Mintz et al., 2009; Nguyen and Moschitti, 2011). Nothman et al. (2013) automatically created massive, multilingual training annotations for NER by exploiting the text and internal structure of Wikipedia. They first categorised Wikipedia articles into a specific set of named entity types across nine languages: Dutch, English, French, German, Italian, Polish, Portuguese, Rus-

sian, and Spanish. Then, Wikipedia's links were transformed into named entity annotations based on the NE types of the target articles. Following this approach, millions of words were annotated in the aforementioned nine languages. Their method for automatically deriving corpora from Wikipedia outperformed the methods proposed by Richman and Schone (2008) and Mika et al. (2008) when testing the Wikipedia-trained models on CONLL shared task data and other gold-standard corpora. Alotaibi and Lee (2013) presented a methodology to automatically build two NE-annotated sets from Arabic Wikipedia. The corpora were built by transforming links into NE annotations according to the NE type of the target articles. POS-tagging, morphological analysis, and linked NE phrases were used to detect other mentions of NEs that appear without links in text. Their Wikipedia-trained model performed well when tested on various newswire test sets, but it did not surpass the performance of the supervised classifier that is trained and tested on data sets drawn from the same domain.

2.3 Classifier Combination and NER

We are not aware of any previous work combining minimally supervised methods for NER task in Arabic or any other natural language, but there are many studies that have examined classifier combination schemes to combine various supervised-learning systems. Florian et al. (2003) presented the best system at the NER CoNLL 2003 task, with an F-score value equal to 88.76%. They used a combination of four diverse NE classifiers: the transformation-based learning classifier, a Hidden Markov Model classifier (HMM), a robust risk minimization classifier based on a regularized winnow method (Zhang et al., 2002), and a ME classifier. The features they used included tokens, POS and chunk tags, affixes, gazetteers, and the output of two other NE classifiers trained on richer datasets. Their methods for combining the results of the four NE classifiers improved the overall performance by 17-21% when compared with the best performing classifier.

Saha and Ekbal (2013) studied classifier combination techniques for various NER models under single and multi-objective optimisation frameworks. They used seven diverse classifiers - naive Bayes, decision tree, memory based learner, HMM, ME, CRFs, and SVMs - to build a number of voting models based on identified text features that are selected mostly without domain knowledge. The combination methods used were binary and real vote-based ensembles. They reported that the proposed multi-objective optimisation classifier ensemble with real voting outperforms the individual classifiers, the three baseline ensembles, and the corresponding single objective classifier ensemble.

3 Two Minimally Supervised NER Classifiers

Two main minimally supervised approaches have been used for NER: semi-supervised learning (Althobaiti et al., 2013) and distant supervision (Nothman et al., 2013). We developed state-of-the-art classifiers of both types that will be used as base classifiers in this paper. Our implementations of these classifiers are explained in Section 3.1 and Section 3.2.

3.1 Semi-supervised Learning

As previously mentioned, the most common SSL technique is bootstrapping, which only requires a set of seeds to initiate the learning process. We used an algorithm adapted from Althobaiti et al. (2013) and contains three components, as shown in Figure 1.

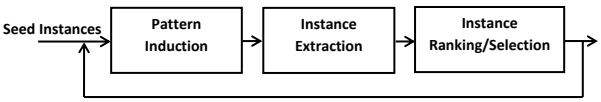

Figure 1: The Three Components of SSL System.

The algorithm begins with a list of a few examples of a given NE type (e.g., 'London' and 'Paris' can be used as seed examples for location entities) and learns patterns *(P)* that are used to find more examples (candidate NEs). These examples are eventually sorted and used again as seed examples for the next iteration.

Our algorithm does not use plain frequencies since absolute frequency does not always produce good examples. This is because bad examples will be extracted by one pattern, however unwantedly, as many times as the bad examples appear in the text in relatively similar contexts. Meanwhile, good exam-

ples are best extracted using more than one pattern, since they occur in a wider variety of contexts in the text. Instead, our algorithm ranks candidate NEs according to the number of different patterns that are used to extract them, since pattern variety is a better cue to semantics than absolute frequency (Baroni et al., 2010).

After sorting the examples according to the number of distinct patterns, all examples but the top m are discarded, where m is set to the number of examples from the previous iteration, plus one. These m examples will be used in the next iteration, and so on. For example, if we start the algorithm with 20 seed instances, the following iteration will start with 21, and the next one will start with 22, and so on. This procedure is necessary in order to carefully include examples from one iteration to another and to ensure that bad instances are not passed on to the next iteration. The same procedure was applied by (Althobaiti et al., 2013).

3.2 Distant Learning

For distant learning we follow the state of the art approach to exploit Wikipedia for Arabic NER, as in (Althobaiti et al., 2014). Our distant learning system exploits many of Wikipedia's features, such as anchor texts, redirects, and inter-language links, in order to automatically develop an Arabic NE annotated corpus, which is used later to train a state-of-the-art supervised classifier. The three steps of this approach are:

1. Classify Wikipedia articles into a set of NE types.
2. Annotate the Wikipedia text as follows:
    - Identify and label matching text in the title and the first sentence of each article.
    - Label linked phrases in the text according to the NE type of the target article.
    - Compile a list of alternative titles for articles and filter out ambiguous ones.
    - Identify and label matching phrases in the list and the Wikipedia text.
3. Filter sentences to prevent noisy sentences from being included in the corpus.

We briefly explain these steps in the following sections.

3.2.1 Classifying Wikipedia Articles

The Wikipedia articles in the dataset need to be classified into the set of named entity types in the classification scheme. We conduct an experiment that uses simple bag-of-words features extracted from different portions of the Wikipedia document and metadata such as categories, the infobox table, and tokens from the article title and first sentence of the document. To improve the accuracy of document classification, tokens are distinguished based on their location in the document. Therefore, categories and infobox features are marked with suffixes to differentiate them from tokens extracted from the article's body text (Tardif et al., 2009). The feature set is represented by Term Frequency-Inverse Document Frequency (*TF-IDF*). In order to develop a Wikipedia document classifier to categorise Wikipedia documents into CoNLL NE types, namely person, location, organisation, miscellaneous, or other, we use a set of 4,000 manually classified Wikipedia articles that are available free online (Alotaibi and Lee, 2012). 80% of the 4,000 hand-classified Wikipedia articles are used for training, and 20% for evaluation. The Wikipedia document classifier that we train performs well, achieving an F-score of 90%. The classifier is then used to classify all Wikipedia articles. At the end of this stage, we obtain a list of pairs containing each Wikipedia article and its NE Type in preparation for the next stage: developing the NE-tagged training corpus.

3.2.2 The Annotation Process

To begin the Annotation Process we identify matching terms in the article title and the first sentence and then tag the matching phrases with the NE-type of the article. The system adopts partial matching where all corresponding words in the title and the first sentence should first be identified. Then, the system annotates them and all words in between (Althobaiti et al., 2014). The next step is to transform the links between Wikipedia articles into NE annotations according to the NE-type of the link target.

Wikipedia also contains a fair amount of NEs without links. We follow the technique proposed by Nothman et al. (2013), which suggests inferring additional links using the aliases for each article.

Thus, we compile a list of alternative titles, including anchor texts and NE redirects (i.e., the linked phrases and redirected pages that refer to NE articles). It is necessary to filter the list, however, to remove noisy alternative titles, which usually appear due to (a) one-word meaningful named entities that are ambiguous when taken out of context and (b) multi-word alternative titles that contain apposition words (e.g., 'President', 'Vice Minister'). To this end we use the filtering algorithm proposed by Althobaiti et al. (2014) (see Algorithm 1). In this algorithm a capitalisation probability measure for Arabic is introduced. This involves finding the English gloss for each one-word alternative name and then computing its probability of being capitalised in the English Wikipedia. In order to find the English gloss for Arabic words, Wikipedia Arabic-to-English cross-lingual links are exploited. In case the English gloss for the Arabic word could not be found using inter-language links, an online translator is used. Before translating the Arabic word, a light stemmer is used to remove prefixes and conjunctions in order to acquire the translation of the word itself without its associated affixes. The capitalisation probability is computed as follows

$$Pr[EN] = \frac{f(EN)_{isCapitalised}}{f(EN)_{isCapitalised}+f(EN)_{notCapitalised}}$$

where EN is the English gloss of the alternative name; $f(EN)_{isCapitalised}$ is the number of times the English gloss EN is capitalised in the English Wikipedia; and $f(EN)_{notCapitalised}$ is the number of times the English gloss EN is not capitalised in the English Wikipedia. By specifying a capitalisation threshold constraint, ambiguous one-word titles are prevented from being included in the list of alternative titles. The capitalisation threshold is set to 0.75 as suggested in (Althobaiti et al., 2014). The multi-word alternative name is also omitted if any of its words belong to the list of apposition words.

3.2.3 Building The Corpus

The last stage is to incorporate sentences into the final corpus. We refer to this dataset as the Wikipedia-derived corpus (WDC). It contains 165,119 sentences of around 6 million tokens. Our model was then trained on the WDC corpus. In this paper we refer to this model as the DL classifier.

Algorithm 1: Filtering Alternative Names

Input: A set $L = \{l_1, l_2, \ldots, l_n\}$ of all alternative names of Wikipedia articles
Output: A set $RL = \{rl_1, rl_2, \ldots, rl_n\}$ of reliable alternative names

```
for i ← 1 to n do
    T ← split l_i into tokens
    if (T.size() >= 2) then
        /* All tokens of T do not belong to
           apposition list                    */
        if (! containAppositiveWord(T)) then
            add l_i to the set RL
    else
        light_stem ← findLightStem(l_i)
        english_gloss ← translate(light_stem)
        /* Compute Capitalisation
           Probability for English gloss      */
        cap_prob ← compCapProb(english_gloss)
        if (cap_prob > 0.75) then
            add l_i to the set RL
```

The WDC dataset is available online[1]. We also plan to make the models available to the research community.

4 Classifier Combination

4.1 The Case for Classifier Combination

In what follows we use SSL to refer to our semi-supervised classifier (see Section 3.1) and DL to refer to our distant learning classifier (see Section 3.2). Table 1 shows the results of both classifiers when tested on the ANERcorp test set (see Section 5 for details about the dataset).

NEs	Classifiers	Precision	Recall	$F_{\beta=1}$
PER	SSL	85.91	51.10	64.08
	DL	80.01	45.11	57.69
LOC	SSL	87.91	62.48	73.04
	DL	75.21	67.14	70.95
ORG	SSL	84.27	40.30	54.52
	DL	74.10	57.02	64.45
Overall	SSL	86.03	51.29	64.27
	DL	76.44	56.42	64.92

Table 1: The results of SSL and DL classifiers on the ANERcorp test set.

As is apparent in Table 1, the SSL classifier tends to be more precise at the expense of recall. The dis-

[1] https://sites.google.com/site/mahajalthobaiti/resources

tant learning technique is lower in precision than the semi-supervised learning technique, but higher in recall. Generally, preference is given to the distant supervision classifier in terms of F-score.

The classifiers have different strengths. Our semi-supervised algorithm iterates between pattern extraction and candidate NEs extraction and selection. Only the candidate NEs that the classifier is most confident of are added at each iteration, which results in the high precision. The SSL classifier performs better than distant learning in detecting NEs that appear in *reliable/regular patterns*. These patterns are usually learned easily during the training phase, either because they contain important NE indicators[2] or because they are supported by many reliable candidate NEs. For example, the SSL classifier has a high probability to successfully detect اوباما "Obama" and لويس فان خال "Louis van Gaal" as person names in the following sentences:

- صرّح الرئيس اوباما الذي يزور بريطانيا ... "President Obama said on a visit to Britain ..."
- قال لويس فان خال مدرب مانشستر يونايتيد ان ... "Louis van Gaal the manager of Manchester United said that ..."

The patterns extracted from such sentences in the newswire domain are learned easily during the training phase, as they contain good NE indicators like الرئيس "president" and مدرب "manager".

Our distant learning method relies on Wikipedia structure and links to automatically create NE annotated data. It also depends on Wikipedia features, such as inter-language links and redirects, to handle the rich morphology of Arabic without the need to perform excessive pre-processing steps (e.g., POS-tagging, deep morphological analysis), which has a slight negative effect on the precision of the DL classifier. The recall, however, of the DL classifier is high, covering as many NEs as possible in all possible domains. Therefore, the DL classifier is better than the SSL classifier in detecting NEs that appear in ambiguous contexts (they can be used for different NE types) and with no obvious clues (NE indicators). For example, detecting فيراري"Ferrari" and نوكيا"Nokia" as organization names in the following sentences:

- تقدم الونسو على سائق رينو، الذي حرم فيراري ... "Alonso got ahead of the Renault driver who prevented Ferrari from ... "
- جاء خطاب نوكيا بعد يوم من اعلان اتمام الصفقة "Nokia's speech came a day after the completion of the deal"

The strengths and weaknesses of the SSL and DL classifiers indicates that a classifier ensemble could perform better than its individual components.

4.2 Classifier Combination Methods

Classifier combination methods are suitable when we need to make the best use of the predictions of multiple classifiers to enable higher accuracy classifications. Dietterich (2000a) reviews many methods for constructing ensembles and explains why classifier combination techniques can often gain better performance than any base classifier. Tulyakov et al. (2008) introduce various categories of classifier combinations according to different criteria including the type of the classifier's output and the level at which the combinations operate. Several empirical and theoretical studies have been conducted to compare ensemble methods such as boosting, randomisation, and bagging techniques (Maclin and Opitz, 1997; Dietterich, 2000b; Bauer and Kohavi, 1999). Ghahramani and Kim (2003) explore a general framework for a Bayesian model combination that explicitly models the relationship between each classifier's output and the unknown true label. As such, multiclass Bayesian Classifier Combination (BCC) models are developed to combine predictions of multiple classifiers. Their proposed method for BCC in the machine learning context is derived directly from the method proposed in (Haitovsky et al., 2002) for modelling disagreement between human assessors, which in turn is an extension of (Dawid and Skene, 1979). Similar studies for modelling data annotation using a variety of methods are presented in (Carpenter, 2008; Cohn and Specia, 2013). Simpson et al. (2013) present a variant of BCC in which they consider the use of a principled approximate Bayesian method, variational Bayes (VB), as an inference technique instead of using Gibbs Sampling. They also alter the model so as to use point values for hyper-parameters, instead of placing exponential hyper-priors over them.

[2] Also known as trigger words which help in identifying NEs within text

The following sections detail the combination methods used in this paper to combine the minimally supervised classifiers for Arabic NER.

4.2.1 Voting

Voting is the most common method in classifier combination because of its simplicity and acceptable results (Van Halteren et al., 2001; Van Erp et al., 2002). Each classifier is allowed to vote for the class of its choice. It is common to take the majority vote, where each base classifier is given one vote and the class with the highest number of votes is chosen. In the case of a tie, when two or more classes receive the same number of votes, a random selection is taken from among the winning classes. It is useful, however, if base classifiers are distinguished by their quality. For this purpose, weights are used to encode the importance of each base classifier (Van Erp et al., 2002).

Equal voting assumes that all classifiers have the same quality (Van Halteren et al., 2001). Weighted voting, on the other hand, gives more weight to classifiers of better quality. So, each classifier is weighted according to its overall precision, or its precision and recall on the class it suggests.

Formally, given *K* classifiers, a widely used combination scheme is through the linear interpolation of the classifiers' class probability distribution as follows

$$P(C|\mathrm{S}_1^K(w)) = \sum_{k=1}^{K} P_k(C|S_k(w)) \cdot \lambda_k(w)$$

where $P_k(C|S_k(w))$ is an estimation of the probability that the correct classification is *C* given $S_k(w)$, the class for the word *w* as suggested by classifier *k*. $\lambda_k(w)$ is the weight that specifies the importance given to each classifier *k* in the combination.

$P_k(C|S_k(w))$ is computed as follows

$$P_k(C|S_k(w)) = \begin{cases} 1, & \text{if } S_k(w) = C \\ 0, & \text{otherwise} \end{cases}$$

For equal voting, each classifier should have the same weight (e.g., $\lambda_k(w) = 1/K$). In case of weighted voting, the weight associated with each classifier can be computed from its precision and/or recall as illustrated above.

4.2.2 Independent Bayesian Classifier Combination (IBCC)

Using a Bayesian approach to classifier combination (BCC) provides a mathematical combination framework in which many classifiers, with various distributions and training features, can be combined to provide more accurate information. This framework explicitly models the relationship between each classifier's output and the unknown true label (Levenberg et al., 2014). This section describes the Bayesian approach to the classifier combination we adopted in this paper which, like the work of Levenberg et al. (2014), is based on Simpson et al. (2013) simplification of Ghahramani and Kim (2003) model.

For *ith* data point, true label t_i is assumed to be generated by a multinomial distribution with the parameter δ: $p(t_i = j|\delta) = \delta_j$, which models the class proportions. True labels may take values $t_i = 1...J$, where *J* is the number of true classes. It is also assumed that there are K base classifiers. The output of the classifiers are assumed to be discrete with values $l = 1...L$, where *L* is the number of possible outputs. The output $c_i^{(k)}$ of the classifier *k* is assumed to be generated by a multinomial distribution with parameters $\pi_j^{(k)} : p(c_i^{(k)} = l|t_i = j, \pi_j^{(k)}) = \pi_{j,l}^{(k)}$ where $\pi^{(k)}$ is the confusion matrix for the classifier *k*, which quantifies the decision-making abilities of each base classifier.

As in Simpson et al. (2013) study, we assume that parameters $\pi_j^{(k)}$ and δ have Dirichlet prior distributions with hyper-parameters $\alpha_{0,j}^{(k)} = [\alpha_{0,j1}^{(k)}, \alpha_{0,j2}^{(k)}, ..., \alpha_{0,jL}^{(k)}]$ and $\nu = [\nu_{0,1}, \nu_{0,2}, ..., \nu_{0,J}]$ respectively. Given the observed class labels and based on the above prior, the joint distribution over all variables for the IBCC model is

$$p(\delta, \Pi, t, c|A_0, \nu) = \prod_{i=1}^{I} \{\delta_{t_i} \prod_{k=1}^{K} \pi_{t_i, c_i^{(k)}}^{(k)}\} p(\delta|\nu) p(\Pi|A),$$

where $\Pi = \{\pi_j^{(k)}|j = 1...J, k = 1...K\}$ and $A_0 = \{\alpha_{0,j}^{(k)}|j = 1...J, k = 1...K\}$. The conditional probability of a test data point t_i being assigned class *j* is given by

$$p(t_i = j) = \frac{\rho_{ij}}{\sum_{y=1}^{J} \rho_{iy}},$$

where

$$\rho_{ij} = \delta_j \prod_{k=1}^{K} \pi_{j,c_i^{(k)}}.$$

In our implementation we used point values for A_0 as in (Simpson et al., 2013). The values of hyper-parameters A_0 offered a natural method to include any prior knowledge. Thus, they can be regarded as pseudo-counts of prior observations and they can be chosen to represent any prior level of uncertainty in the confusion matrices, Π. Our inference technique for the unknown variables (δ, π, and t) was Gibbs sampling as in (Ghahramani and Kim, 2003; Simpson et al., 2013). Figure 2 shows the directed graphical model for IBCC. The $c_i^{(k)}$ represents observed values, circular nodes are variables with distributions and square nodes are variables instantiated with point values.

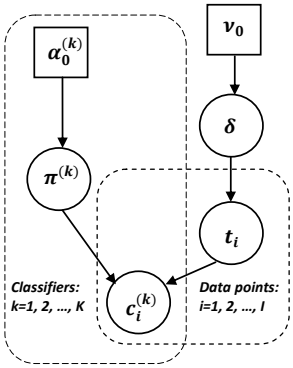

Figure 2: The directed graph of IBCC.

5 Data

In this section, we describe the two datasets we used:

- Validation set[3](NEWS + BBCNEWS): 90% of this dataset is used to estimate the weight of each base classifier and 10% is used to perform error analysis.
- Test set (ANERcorp test set): This dataset is used to evaluate different classifier combination methods.

The validation set is composed of two datasets: NEWS and BBCNEWS. The NEWS set contains around 15k tokens collected by Darwish (2013) from the RSS feed of the Arabic (Egypt) version of news.google.com from October 2012. We created the BBCNEWS corpus by collecting a representative sample of news from BBC in May 2014. It contains around 3k tokens and covers different types of news such as politics, economics, and entertainment.

The ANERcorp test set makes up 20% of the whole ANERcorp set. The ANERcorp set is a newswire corpus built and manually tagged especially for the Arabic NER task by Benajiba et al. (2007a) and contains around 150k tokens. This test set is commonly used in the Arabic NER literature to evaluate supervised classifiers (Benajiba and Rosso, 2008; Abdul-Hamid and Darwish, 2010; Abdallah et al., 2012; Oudah and Shaalan, 2012) and minimally-supervised classifiers (Alotaibi and Lee, 2013; Althobaiti et al., 2013; Althobaiti et al., 2014), which allows us to review the performance of the combined classifiers and compare it to the performance of each base classifier.

[3] Also known as development set.

6 Experimental Analysis

6.1 Experimental Setup

In the IBCC model, the validation data was used as known t_i to ground the estimates of model parameters. The hyper-parameters were set as $\alpha_j^{(k)} = 1$ and $\nu_j = 1$ (Kim and Ghahramani, 2012; Levenberg et al., 2014). The initial values for random variables were set as follows: (a) the class proportion δ was initialised to the result of counting t_i and (b) the confusion matrix π was initialised to the result of counting t_i and the output of each classifier $c^{(k)}$. Gibbs sampling was run well past stability (i.e., 1000 iterations). Stability was actually reached in approximately 100 iterations.

All parameters required in voting methods were specified using the validation set. We examined two different voting methods: equal voting and weighted voting. In the case of equal voting, each classifier was given an equal weight, (1/K) where K was the number of classifiers to be combined. In weighted voting, total precision was used in order to give preference to classifiers with good quality.

6.2 Results and Discussion

6.2.1 A Simple Baseline Combined Classifier

A proposed combined classifier simply and straightforwardly makes decisions based on the agreed decisions of the base classifiers, namely the SSL classifier and DL classifier. That is, if the base classifiers agree on the NE type of a certain word, then it is annotated by an agreed NE type. In the case of disagreement, the word is considered *not* named entity. Table 2 shows the results of this combined classifier, which is considered a baseline in this paper.

	Precision	**Recall**	$\mathbf{F}_{\beta=1}$
Person	97.31	24.69	39.39
Location	98.35	40.01	56.88
Organisation	97.38	33.2	49.52
Overall	97.68	32.63	48.92

Table 2: The results of the baseline

The results of the combined classifier shows very high precision, which indicates that both base classifiers are mostly accurate. The base classifiers also commit different errors that are evident in the low recall. The accuracy and diversity of the single classifiers are the main conditions for a combined classifier to have better accuracy than any of its components (Dietterich, 2000a). Therefore, in the next section we take into consideration various classifier combination methods in order to aggregate the best decisions of SSL and DL classifiers, and to improve overall performance.

6.2.2 Combined Classifiers: Classifier Combination Methods

The SSL and DL classifiers are trained with two different algorithms using different training data. The SSL classifier is trained on ANERcorp training data, while the DL classifier is trained on a corpus automatically derived from Arabic Wikipedia, as explained in Section 3.1 and 3.2.

We combine the SSL and DL classifiers using the three classifier combination methods, namely equal voting, weighted voting, and IBCC. Table 3 shows the results of these classifier combination methods. The IBCC scheme outperforms all voting techniques and base classifiers in terms of F-score. Regarding precision, voting techniques show the highest scores. However, the high precision is accompanied by a reduction in recall for both voting methods. The IBCC combination method also has relatively high precision compared to the precision of base classifiers. Much better recall is registered for IBCC, but it is still low.

NEs	**Combination Methods**	**Precision**	**Recall**	$\mathbf{F}_{\beta=1}$
PER	Equal Voting	79.99	41.88	54.97
	Weighted Voting	80.15	44.24	57.01
	IBCC	77.87	63.86	70.17
LOC	Equal Voting	86.87	30.66	45.32
	Weighted Voting	87.48	30.23	44.93
	IBCC	81.52	59.86	69.03
ORG	Equal Voting	97.01	29.97	45.79
	Weighted Voting	98.11	30.98	47.09
	IBCC	95.44	34.31	50.47
Overall	Equal Voting	87.96	34.17	49.22
	Weighted Voting	88.58	35.15	50.33
	IBCC	84.94	52.68	65.03
NEs	**Base Classifiers**	**Precision**	**Recall**	$\mathbf{F}_{\beta=1}$
Overall	SSL	86.03	51.29	64.27
	DL	76.44	56.42	64.92

Table 3: The performances of various combination methods.

6.2.3 Combined Classifiers: Restriction of the Combination Process

An error analysis of the validation set shows that 10.01% of the NEs were correctly detected by the semi-supervised classifier, but considered *not* NEs by the distant learning classifier. At the same time, the distant learning classifier managed to correctly detect 25.44% of the NEs that were considered *not* NEs by the semi-supervised classifier. We also noticed that false positive rates, i.e. the possibility of considering a word NE when it is actually *not* NE, are very low (0.66% and 2.45% for the semi-supervised and distant learning classifiers respectively). These low false positive rates and the high percentage of the NEs that are detected and missed by the two classifiers in a mutually exclusive way can be exploited to obtain better results, more specifically, to increase recall without negatively affecting precision. Therefore, we restricted the combi-

nation process to only include situations where the base classifiers agree or disagree on the NE type of a certain word. The combination process is ignored in cases where the base classifiers only disagree on detecting NEs. For example, if the base classifiers disagree on whether a certain word is an NE or not, the word is automatically considered an NE. Figure 3 provides some examples that illustrate the restrictions we applied to the combination process. The annotations in the examples are based on the CoNLL 2003 annotation guidelines (Chinchor et al., 1999).

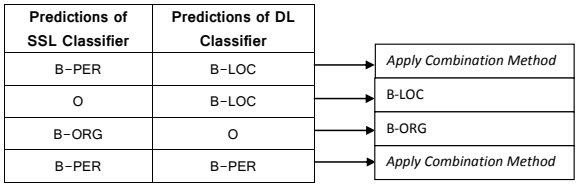

Figure 3: Examples of restricting the combination process.

Restricting the combination process in this way increases recall without negatively affecting the precision, as seen in Table 4. The increase in recall makes the overall F-score for all combination methods higher than those of base classifiers. This way of using the IBCC model results in a performance level that is superior to all of the individual classifiers and other voting-based combined classifiers. Therefore, the IBCC model leads to a 12% increase in the performance of the best base classifier, while voting methods increase the performance by around 7% - 10%. These results highlight the role of restricting the combination, which affects the performance of combination methods and gives more control over how and when the predictions of base classifiers should be combined.

6.2.4 Comparing Combined Classifiers: Statistical Significance of Results

We tested whether the difference in performance between the three classifier combination methods - equal voting, weighted voting, and IBCC - is significant using two different statistical tests over the results of these combination methods on an ANERcorp test set. The alpha level of 0.01 was used as a significance criterion for all statistical tests. First, We ran a non-parametric sign test. The small p-value ($p \ll 0.01$) for each pair of the three combination methods, as seen in Table 5, suggests that these methods are significantly different. The only comparison where no significance was found is *equal voting* vs. *weighted voting*, when we used them to combine the data without any restrictions ($p = 0.3394$).

NEs	Combination Methods	Precision	Recall	$F_{\beta=1}$
PER	Equal Voting	74.46	61.88	67.59
	Weighted Voting	77.77	63.50	69.91
	IBCC	77.88	64.56	**70.60**
LOC	Equal Voting	74.04	71.36	72.68
	Weighted Voting	74.05	73.70	73.86
	IBCC	76.20	75.91	**76.05**
ORG	Equal Voting	76.01	63.97	69.47
	Weighted Voting	76.30	66.60	71.12
	IBCC	78.91	66.65	**72.26**
Overall	Equal Voting	74.84	65.74	69.99
	Weighted Voting	76.04	67.93	71.76
	IBCC	77.66	69.04	**73.10**
NEs	**Base Classifiers**	**Precision**	**Recall**	$F_{\beta=1}$
Overall	SSL	86.03	51.29	64.27
	DL	76.44	56.42	64.92

Table 4: The performances of various combination methods when restricting the combination process.

Combination Methods (Without Restriction)			
	Equal Voting	Weighted Voting	IBCC
Equal Voting			
Weighted Voting	0.3394		
IBCC	<2.2E-16	<2.2E-16	
Combination Methods (With Restriction)			
	Equal Voting	Weighted Voting	IBCC
Equal Voting			
Weighted Voting	1.78E-07		
IBCC	<2.2E-16	1.97E-06	

Table 5: The sign test results (exact p values) for the pairwise comparisons of the combination methods.

Second, we used a bootstrap sampling (Efron and Tibshirani, 1994), which is becoming the de facto standards in NLP (Søgaard et al., 2014). Table 6 compares each pair of the three combination methods using a bootstrap sampling over documents with 10,000 replicates. It shows the p-values and confidence intervals of the difference between means.

Combination With Restriction		
Combination Methods Comparison	p-value	[95% CI]
Weighted Voting, Equal Voting	0.000	[0.270, 0.600]
IBCC, Equal Voting	0.000	[0.539, 0.896]
IBCC, Weighted Voting	0.000	[0.157, 0.426]
Combination Without Restriction		
Combination Methods Comparison	p-value	[95% CI]
Weighted Voting, Equal Voting	0.508	[-0.365, 0.349]
IBCC, Equal Voting	0.000	[4.800, 6.122]
IBCC, Weighted Voting	0.000	[4.783, 6.130]

Table 6: The bootstrap test results (p-values and CI) for the pairwise comparisons of the combination methods.

The differences in performance between almost all the three methods of combination are highly significant. The one exception is the comparison between *equal voting* and *weighted voting*, when they are used as a combination method without restriction, which shows a non-significant difference (p-value = 0.508, CI = -0.365 to 0.349).

Generally, the IBCC scheme performs significantly better than voting-based combination methods whether we impose restrictions on the combination process or not, as can be seen in Table 3 and Table 4.

7 Conclusion

Major advances over the past decade have occurred in Arabic NER with regard to utilising various supervised systems, exploring different features, and producing manually annotated corpora that mostly cover the standard set of NE types. More effort and time for additional manual annotations are required when expanding the set of NE types, or exporting NE classifiers to new domains. This has motivated research in minimally supervised methods, such as semi-supervised learning and distant learning, but the performance of such methods is lower than that achieved by supervised methods. However, semi-supervised methods and distant learning tend to have different strengths, which suggests that better results may be obtained by combining these methods. Therefore, we trained two classifiers based on distant learning and semi-supervision techniques, and then combined them using a variety of classifier combination schemes. Our main contributions include the following:

- We presented a novel approach to Arabic NER using a combination of semi-supervised learning and distant supervision.
- We used the Independent Bayesian Classifier Combination (IBCC) scheme for NER, and compared it to traditional voting methods.
- We introduced the classifier combination restriction as a means of controlling how and when the predictions of base classifiers should be combined.

This research demonstrated that combining the two minimal supervision approaches using various classifier combination methods leads to better results for NER. The use of IBCC improves the performance by 8 percentage points over the best base classifier, whereas the improvement in the performance when using voting methods is only 4 to 6 percentage points. Although all combination methods result in an accurate classification, the IBCC model achieves better recall than other traditional combination methods. Our experiments also showed how restricting the combination process can increase the recall ability of all the combination methods without negatively affecting the precision.

The approach we proposed in this paper can be easily adapted to new NE types and different domains without the need for human intervention. In addition, there are many ways to restrict the combination process according to the applications' preferences, either producing high accuracy or recall. For example, we may obtain a highly accurate combined classifier if we do not combine the predictions of all base classifiers for a certain word and automatically consider it *not* NE when one of the base classifier considers this word *not* NE.

References

Sherief Abdallah, Khaled Shaalan, and Muhammad Shoaib. 2012. Integrating rule-based system with classification for arabic named entity recognition. In *Computational Linguistics and Intelligent Text Processing*, pages 311–322. Springer.

Samir AbdelRahman, Mohamed Elarnaoty, Marwa Magdy, and Aly Fahmy. 2010. Integrated machine

learning techniques for arabic named entity recognition. *IJCSI*, 7:27–36.

Ahmed Abdul-Hamid and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115. Association for Computational Linguistics.

Steven Abney. 2010. *Semisupervised learning for computational linguistics*. CRC Press.

Fahd Alotaibi and Mark Lee. 2012. Mapping Arabic Wikipedia into the named entities taxonomy. In *Proceedings of COLING 2012: Posters*, pages 43–52, Mumbai, India, December. The COLING 2012 Organizing Committee.

Fahd Alotaibi and Mark Lee. 2013. Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia. In *IJCNLP*.

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2013. A semi-supervised learning approach to arabic named entity recognition. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 32–40, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2014. Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 106–115, Gothenburg.

Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A Corpus-Based Semantic Model Based on Properties and Types. *Cognitive Science*, 34(2):222–254.

Eric Bauer and Ron Kohavi. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139.

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007a. Anersys: An Arabic Named Entity Recognition System based on Maximum Entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153. Springer.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007b. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153. Springer.

Yassine Benajiba, Mona Diab, Paolo Rosso, et al. 2008. Arabic named entity recognition: An svm-based approach. In *Proceedings of 2008 Arab International Conference on Information Technology (ACIT)*, pages 16–18.

Bob Carpenter. 2008. Multilevel bayesian models of categorical data annotation. Unpublished manuscript. Available online at http://lingpipe-blog.com/lingpipe-white-papers/, last accessed 15-March-2015.

Nancy Chinchor, Erica Brown, Lisa Ferro, and Patty Robinson. 1999. 1999 Named Entity Recognition Task Definition. *MITRE and SAIC*.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *ACL*, pages 32–42.

Kareem Darwish. 2013. Named Entity Recognition using Cross-lingual Resources: Arabic as an Example. In *ACL*, pages 1558–1567.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28.

Thomas G. Dietterich. 2000a. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg.

Thomas G Dietterich. 2000b. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Ali Elsebai, Farid Meziane, and Fatma Zohra Belkredim. 2009. A rule based persons names arabic extraction system. *Communications of the IBIMA*, 11(6):53–59.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171. Association for Computational Linguistics.

Zoubin Ghahramani and Hyun-Chul Kim. 2003. Bayesian classifier combination. Technical report, University College London.

Y Haitovsky, A Smith, and Y Liu. 2002. Modelling disagreements among and within raters assessments from the bayesian point of view. In *Draft. Presented at the Valencia meeting 2002*.

Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *International conference on artificial intelligence and statistics*, pages 619–627.

Abby Levenberg, Stephen Pulman, Karo Moilanen, Edwin Simpson, and Stephen Roberts. 2014. Predicting economic indicators from web text using sentiment composition. *International Journal of Computer and Communication Engineering*, 3(2):109–115.

Richard Maclin and David Opitz. 1997. An empirical evaluation of bagging and boosting. *AAAI/IAAI*, 1997:546–551.

Slim Mesfar. 2007. Named entity recognition for arabic using syntactic grammars. In *Natural Language Processing and Information Systems*, pages 305–316. Springer.

Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. 2008. Learning to Tag and Tagging to Learn: A Case Study on Wikipedia. volume 23, pages 26–33.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

David Nadeau. 2007. Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision.

Truc-Vien T. Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 277–282, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual Named Entity Recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Mai Oudah and Khaled F Shaalan. 2012. A pipeline arabic named entity recognition using a hybrid approach. In *COLING*, pages 2159–2176.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405.

Alexander E Richman and Patrick Schone. 2008. Mining Wiki Resources for Multilingual Named Entity Recognition. In *ACL*, pages 1–9.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI*, pages 474–479.

Sriparna Saha and Asif Ekbal. 2013. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, 85:15–39.

Satoshi Sekine et al. 1998. NYU: Description of the Japanese NE system used for MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, volume 17.

Khaled Shaalan and Hafsa Raza. 2009. Nera: Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology*, 60(8):1652–1663.

Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, pages 1–35. Springer.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Hector Martinez. 2014. Whats in a p-value in nlp? In *Proceedings of the eighteenth conference on computational natural language learning (CONLL14)*, pages 1–10.

Sam Tardif, James R. Curran, and Tara Murphy. 2009. Improved Text Categorisation for Wikipedia Named Entities. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 104–108.

Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann. 2008. Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition*, pages 361–386. Springer.

Merijn Van Erp, Louis Vuurpijl, and Lambert Schomaker. 2002. An overview and comparison of voting methods for pattern recognition. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 195–200. IEEE.

Hans Van Halteren, Walter Daelemans, and Jakub Zavrel. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational linguistics*, 27(2):199–229.

Wajdi Zaghouani. 2014. Critical Survey of the Freely Available Arabic Corpora. In *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*, pages 1–8, Reykjavik, Iceland.

Tong Zhang, Fred Damerau, and David Johnson. 2002. Text chunking based on a generalization of winnow. *The Journal of Machine Learning Research*, 2:615–637.