The IBM Systems for Entity Discovery and Linking at TAC 2017

Avirup Sil and **Georgiana Dinu** and **Gourab Kundu** and **Radu Florian**
IBM T.J. Watson Research Center
1101 Kitchawan Rd
Yorktown Heights, NY 10598
{avi,gdinu,gkundu,raduf}@us.ibm.com

1 Mention Detection

The IBM mention detection system was a combination of two mention detection systems - a Neural Net-based (NN) system and a Conditional Random Fields (CRF) system, both trained to predict the standard IOB mention detection encoding.

CRF Model The CRF model is a linear-chain CRF model of size 1 (the previous tag is used in features), using a multitude of features including words in context, capitalization flags, various entity dictionaries, both supervised (lists extracted from the ACE'05 data, the CoNLL'03 data, etc) and unsupervised (the system output on Gigaword), word clustering (Brown clusters), cache features, word length and IDF.

Neural Network Model The NN system uses a feed-forward neural net to predict entity labels. The network architecture (Figure 1) is similar to that proposed in (Collobert et al., 2011) and uses as input the concatenation of the target and context words (symmetric window of size 4), 1 hidden layer with 1000 hidden units and sigmoid activation. We introduce a recurrence element in the form of embedding the two previously assigned labels as features for the current instance. Note that this is simpler than modeling the label transition probability specifically, as the CRF-objective sequence labeling neural networks. For all languages we use the following additional features:

- Character-level representations: bidirectional LSTMs for English and Spanish and averaged pre-trained character embeddings for Chinese.
- Gazetteers[1] (automatically extracted)
- The output of a mention detection model trained on news-domain in-house data and consisting of 50 mention types.

The word vectors are initialized with 300-dimensional pre-trained embeddings build on a concatenation of crawl, Gigaword and Wikipedia data. Embeddings are built using a variant of the word2vec CBOW architecture, which predicts a target word from the *concatenation* of its context words, rather than the average. This variant outperforms CBOW both on standard word similarity benchmarks as well as in mention detection experiments.

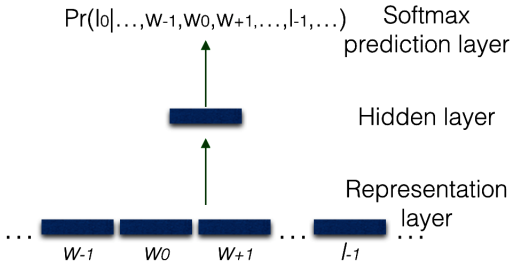

Figure 1: Architecture of the feed forward neural network used for mention detection. Input layer consists of word in context, previous label embeddings, and embeddings of additional features (listed above).

System combination The NN and CRF models have high-precision low-recall which led us to a

[1]We used the same English-language gazetteers for all three languages

	Val-NN	Val-CRF	Tst-NN+CRF
Eng	0.843	0.803	0.806
Spa	0.809	0.785	0.785
Cmn	0.843	0.811	0.699

Table 1: Mention detection results on validation data (first two columns) and official competition results of combined NN+CRF system (third column).

combination scheme which favors recall: The initial system output is the best performing one and, considering any remaining systems in the order of performance, add the mentions that do not overlap with the combined system. We use NN and CRF in this order for Spanish and Chinese and the NN system only for English.

Nominal mentions In addition to the training data annotating both named and nominals, we also made use of previously released data which contained only named entities. Specifically, we added to this data *nominal* annotations obtained from an NN model trained on the first data set (both names and nominals). We subsequently used the concatenation of the two data sets, containing gold *named* annotations and partially self-trained *nominals*, to train a new model.

2 Coreference Resolution

We use two different coreference systems for this evaluation. For languages with gold standard training data (english, chinese, spanish) available from previous years' evaluations, we train mention pair based coreference model using logistic regression. For other low-resource languages with no available gold standard training data, we train an entity pair based coreference model using neural network, from the english coreference data and use this system on new languages without any retraining.

Logistic Regression Model This model is trained using logistic regression over the set of all (mention, antecedent) pairs from every document. Mentions are first ordered on the basis of mention type (name, nominal and pronoun) and then on the basis of the order in which they appear in the document. The antecedents for each mention are all mentions preceding it in the ordering. For a (mention, antecedent) pair (m_j, m_i),

$$P(y = 1|m_j, m_i) = \frac{1}{1 + e^{-w \cdot \phi(m_j, m_i)}} \quad (1)$$

$\phi(m_j, m_i)$ is a hand engineered feature vector including string match (substring match or full string match), distance (word distance and sentence distance discretized into bins), mention types(name or nominal or pronoun; person or location or organization etc.), acronym, first name mismatch and speaker detection. y is a boolean variable that indicates if the pair (m_j, m_i) is coreferent.

During decoding, all mentions are ordered similar to training and the highest scoring antecedent m_i for each mention m_j is identified. If the score is above a threshold, m_j is merged with the entity containing m_i otherwise m_j forms its new entity.

Neural Network Model Unlike the logistic regression model, the NN model is trained and decoded at the entity pair level. For a pair of entities $E_1 = \{m_i\}$ and $E_2 = \{m_j\}$, a cartesian product of $E_1 \times E_2$ is made and the feature vector ϕ_{ij} for each $(m_i, m_j) \in E_1 \times E_2$ is computed. The features in ϕ_{ij} are similar to the features used in the logistic regression model. Every active feature in ϕ_{ij} is then embedded as a vector in the real space. Let v_{ij} dentote the concatenation of embeddings of all active features in ϕ_{ij}. Embeddings of all features except the words are learned in the training process. Word embeddings are pre-trained, obtained by training monolingual word embeddings and then projecting them to the space of english word embeddings using a technique similar to (Mikolov et al., 2013). v_{ij} is then fed to a feed forward neural network and entity pair level representation is obtained by a sum of (mention, antecedent) embeddings weighted by a scalar for the mention type of the mention.

$$e = \sum_{ij} \rho(m_j)\sigma(W v_{ij} + b) \quad (2)$$

$\rho(m_j)$ is a scalar weight for the mention type (name or nominal or pronoun) for mention m_j. Next, e is fed to a softmax layer with two outputs: yes (coreferent) or no (non-coreferent).

During decoding, this model incrementally merges entities to produce the entity clustering for a document. It starts with all singleton entities where

	MUC	B3	CEAF	CoNLL
Eng	0.90	0.89	0.84	0.87
Spa	0.91	0.92	0.88	0.90
Cmn	0.97	0.96	0.91	0.94

Table 2: Coreference results on the test portion of TAC 15 for three languases (Eng, Spa, Cmn) for model trained on training portion of TAC 15 english coreference data.

each mention belongs to its own entity. Then the model decideds to merge entity pairs until no more merge is possible.

Table 2 shows results on the test portion of TAC 15 for three languases (Eng, Spa, Cmn) for model trained on training portion of TAC 15 english coreference data.

3 Entity Linking Formulation

We define the goal of Entity Linking (EL) as, given a textual mention m and a document D, $m \in D$ and $m, D \in en$, to identify the best link l_i:

$$\hat{l}^{(m)} = \arg\max_j P(l_j^{(m)}|m, D) \quad (3)$$

Since computing $P\left(l_j^{(m)}|m, D\right)$ can be prohibitive over large datasets, we change the problem into computing

$$\hat{l}^{m} = \arg\max_j P(C|m, D, l_j^{(m)}) \quad (4)$$

where C is a Boolean variable that measures how "consistent" the pairs (m, D) and $l_j^{(m)}$ are. As a further simplification, given (m, D), we perform an Information Retrieval (IR)-flavored *fast match* to identify the most likely candidate links $l_{j_1}^{(m)}, \ldots, l_{j_m}^{(m)}$ for the input (m, D), then find the $\arg\max$ over this subset.

In cross-lingual EL, we assume that $m, D \in tr$, where tr is some foreign language like Spanish or Chinese. However, we need to link m to some target link $l_i^{(m)}$, where $l_i^{(m)} \in KB_{en}$.

3.1 Modeling Contexts

We build the sub-networks that encode the representation of query mention m in the given query document D. This representation is then compared with the page embedding of the Wikipedia candidate title (through cosine similarity) and the result is fed into the higher network (Figure 2).

Noting that not the entire document D might be useful for disambiguating m, we choose to represent the mention m based only on the surrounding sentences of m in D, in contrast to (He et al., 2013; Francis-Landau et al., 2016), which chose to use the entire document for modeling. We run CNNs on the sentences and LSTMs on the context windows (4 words before and after the mention) to model fine-grained contexts.

3.2 Cross-Lingual Neural Entity Linking

3.2.1 Neural Model Architecture

The general architecture of our neural EL model is described in Figure 2. Our target is to perform "zero shot learning" (Socher et al., 2013; Palatucci et al., 2009) for cross-lingual EL. Hence, we want to learn a model on English data and use it to decode in any other language, provided we have access to multi-lingual embeddings from English and the target language. We allow the model to compute several similarity/coherence *scores* S (feature abstraction layer): which are several measures of similarity of the context of the mention m in the query document and the context of the candidate link's Wikipedia page, described in details in Section **??**, which are fed to a feed-forward neural layer H with weights W_h, bias b_h, and a *sigmoid* non-linearity. The output of H (denoted as h) is computed according to $h = \sigma(W_h S + b_h)$. The output of the binary classifier $p(C|m, D, l)$ is the softmax over the output of the final feed-forward layer O with weights W_0 and bias b_0. $p(C|m, D, L)$ represents the probability of the output class C taking a value of 1 (correct link) or 0 (incorrect link), and is computed as a 2 dimensional vector and given by:

$$p(C|m, D, l) = softmax(W_0 h + b_0) \quad (5)$$

3.3 LIEL: Language Independent Entity Linking

We also run our LIEL model (Sil and Florian, 2016) on the full dataset which has been described and presented in previous TAC participations. Finally, we combine the output of the NN model with LIEL as

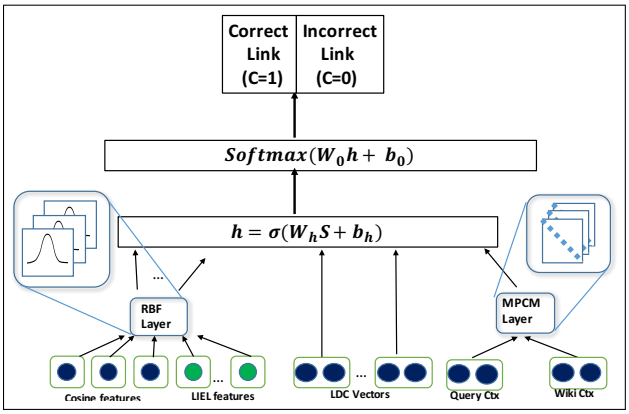

Figure 2: **Architecture of our neural EL system.** The input to the system are: a document D containing the query mention m and the corresponding Wikipedia candidate link $l_i \in L$, where L is the set of all possible links extracted from the fast match step.

Systems	In-KB acc. %
TAC Rank 1	79.2
TAC Rank 2	71.6
Sil & Florian (2016)	78.6
Globerson *et al.* (2016)	87.2
This work	**87.4**

Table 3: **Performance on the TAC 2010 English dataset.**

follows: we start with the NN model and look at the confidence and if the value is less than 70% we choose the output of LIEL.

Some Results We evaluate our proposed method on the benchmark datasets for English: TAC 2010: and Cross-Lingual: TAC 2015 Trilingual Entity Linking dataset.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking

Systems	Linking Acc %
Sil & Florian (2016) / TAC Rank 1	80.4
Tsai & Roth (2016)	80.9
This Work	**81.9**

Table 4: **Performance comparison on the TAC 2015 Spanish dataset.**

Systems	Linking Acc %
TAC Rank 1	83.1
Tsai & Roth (2016)	83.6
This Work	**84.1**

Table 5: **Performance comparison on the TAC 2015 Chinese dataset.**

	NERC	NERLC	CEAFmC
Eng	0.806	0.668	0.713
Spa	0.785	0.603	0.664
Cmn	0.699	0.520	0.593

Table 6: Trilingual EDL results on 2017 TAC Evaluation. Our system obtained the top end-to-end score CEAFmC in English.

with convolutional neural networks. In *Proc. NAACL 2016*.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *ACL (2)*, pages 30–34.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418.

Avirup Sil and Radu Florian. 2016. One for all: Towards language independent named entity linking. *Association for Computational Linguistics*.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.