SUMMA at TAC Knowledge Base Population Task 2017

Afonso Mendes[#] **David Nogueira**[#] **Samuel Broscheit**[#†] **Filipe Aleixo**[#†]
Pedro Balage[#] **Rui Martins**[#] **Sebastião Miranda**[#] **Mariana S. C. Almeida**[#†]
[#]Priberam Labs, Alameda D. Afonso Henriques, 41, 2°, 1000-123 Lisboa, Portugal
[†]Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
{amm,david.nogueira,pedro.balage,sebastiao.miranda}@priberam.pt

Abstract

This paper describes the SUMMA system for the Trilingual Entity Discovery and Linking (EDL) for the TAC 2017 Knowledge Base Population track. We used an entity recognition based on a LSTM+CRF neural network and two different approaches for entity linking disambiguation: a nearest-neighbors search engine and a distributed representation based on previous work of Yamada et al. (2017). We submitted 3 runs for named and nominal EDL task, across all 3 languages (English, Spanish and Chinese). Our system scored as first out of 24 teams for named mention linking (NERLC) and mention clustering (CEAFmC) for English and as third in the same metrics for Spanish.

1 Introduction

Our submission to the NIST TAC-KBP-2017[1] is an attempt to apply our ongoing research on knowledge base population within the SUMMA[2] project to TAC shared tasks. The goal of SUMMA is to develop a scalable and extensible media monitoring platform with an extensible automated knowledge base construction and cross-lingual capabilities, thus having a significant overlap with TAC-KBP tasks. Following last year's submission, restricted to English named entity disambiguation, we present an enhanced version of our Entity Discovery and Linking (EDL) system adapted to the TAC EDL task, both for named and nominal entities, and in the three languages: English, Spanish and Chinese.

The paper is organized as follows: Section 2 describes our contribution to the EDL track. Experimental results are reported in Section 3. Section 4 concludes the paper.

2 Entity Discovery and Linking

2.1 Overview and submissions description

Five systems were submitted to the EDL track, although two of them were discarded as their output did not correlate with our intentions for such runs.

Our entity linking system submitted in run #1, *summa1*, uses an information retrieval Nearest-Neighbors-assisted rule based system (Amaral et al., 2008) to rank candidates and generate additional features from Wikipedia data. Besides nearest-neighbors search engine generated features and prior features, features taken from the co-occurrences between mentions and candidates in Wikipedia, and coherence features (existence or absence of links between Wikipedia articles) are used in the candidates re-ranking steps in this first run.

Runs #4 and #5 follow a different approach for their disambiguation step, as they are anchored in English distributed representations provided by Yamada et al. (2017) for English and trained representations using the same method for Spanish. A slightly different deep neural network from the one described in Yamada et al. (2017) was used in these two runs. Besides distributed representations, run #4 also receives as input data the nearest-neighbors similarity (search engine generated) features, prior and co-occurrences features used in run #1. The underlying idea of these two runs (#4 and #5) was to assess

[1] https://tac.nist.gov//2017/KBP/

[2] http://www.summa-project.eu/

whether these distributed representations captured more relevant information for entity disambiguation than the co-occurrences and link statistics extracted from Wikipedia.

2.2 Entity Recognition

2.2.1 Pre-processing of NER datasets

To improve the results for the named entity recognition, we preprocessed the following corpora that were used for training: TAC 2015 (train); Ontonotes (train); TAC 2016 (dev).

OntoNotes was processed to achieve congruence with the TAC dataset, namely in the name tagging and span boundaries. Specifically, we used the domains of broadcast news, the newswire and the web data; we shortened span boundaries and mapped spans with ”Nationality, or Religious or Political Organization” (NORP) tag to GPE, LOC, ORG or no tag (tag removed).

2.2.2 Named Entity Recognition

The used system is based on the Bi-LSTM+CRF algorithm described by Ma and Hovy (2016)[3], where we used GloVe (Pennington et al., 2014) 300 dimensional pre-trained embeddings for English and we used the word2vec implementation in Gensim (Řehůřek and Sojka, 2010) to train word embeddings on the Spanish Wikipedia, where the anchors have been replaced by the KBID, as described by Yamada et al. (2017).

Regarding ensemble architectures, for English we used an ensemble model from 10 runs with randomly initialized weights; for Spanish, we train a multilingual model using both Spanish and English train data; and for the Chinese submission, we used the Stanford CoreNLP (Manning et al., 2014) as the NER system.

The Spanish model trained only with Spanish TAC dataset reported low scores mainly due the scarce amount of train data available for the language. Our architecture might be described as a jointly trained ensemble and it is illustrated in Figure 1.

In this architecture, English and Spanish submodels share the same architecture. A gating layer is then trained with the goal of linearly combining the logits obtained from the English and Spanish submodels. The gating layer takes as input the list of pre-trained word embeddings for each sentence and averages it, in order to obtain sentence representations. We take the dot product of these sentences and apply a softmax layer that is subsequently used to linearly combine the logits.

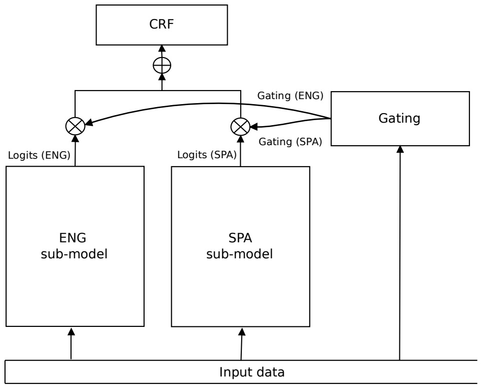

Figure 1: Joint ensemble architecture used for Spanish NER.

The logits obtained from the linear combination are then given as input to a CRF layer, such that a single transitions tensor is learned.

The training procedure is divided into 4 different stages, alternating between English and Spanish and with and without gating between the languages.

2.2.3 Nominal Entity Recognition

For nominal mentions we used a lexicon-based system where the lexicon was constructed from training data and compiled lists. Such lexicon was divided between two extensive lists, one containing possible nominal PER surface mentions and other with GPE, FAC, LOC and ORG surface mentions. In order to create the first list, we compiled people-related nominal mentions concerning professions, family relations and adjectives. Regarding the second list, it contained man-made locations pointing to FAC, natural locations to LOC, locations with some kind of government to GPE and companies or groups of people to ORG. Those lists were the core of the nominal detection system. After being identified in the text, the mentions enter a rule-based system where they are awarded/withdrawn points if they meet specific parameters. The rules take into account the context (previous and following words) as well as part-of-speech (POS) and dependency parsing

[3] Using as starting point the TensorFlow implementation in https://github.com/guillaumegenthial/sequence_tagging

features, extracted using TurboParser[4]. More points were awarded if TurboParser classified the word as a noun and deducted if it was classified as a verb, for instance. The same principle was applied to the dependency relation that the word in question had in the sentence, once again awarding more points if the word was the subject of the sentence. In the case of the word being identified in one of the lists as a FAC or ORG, the article used immediately before was taken into consideration, awarding more points if the article was definite and deducting if it was indefinite. Such system was only used in English and Spanish submission, whereas no Chinese nominal detection was performed.

2.2.4 NER post-processing step

As we used the same previously described entity recognition system on each submission, $summa1$, $summa4$ and $summa5$ share the same mentions list as input for the entity linking step. As a post-processing step, we apply a string matching procedure to capture named mentions that were not recognized by the entity recognition system. In particular, we extract mentions with the exact same surface as those previously detected in the document. These new mentions are then tagged with the types of the old ones, according to a voting procedure that is biased towards the PER label. Later, in Section 2.3, some of the mentions types are also reassigned in order to promote label agreement between mentions in the same mention cluster (after both the coreference and the linking steps).

2.3 Entity Linking

In this section, we describe the multiple approaches undertaken towards named and nominal entity disambiguation. Run #1, $summa1$, is anchored in a nearest-neighbors-assisted ruled-based entity linking system, of which an earlier version was submitted to TAC KBP EDL 2016 (Paikens et al., 2017). The mentions detected as reported in Section 2.2 are linked to KB entries according to the rule-based strategy that will be described in the following sections and summarized in Algorithm 1.

Run #4, $summa4$, of which steps are summarized in Algorithm 2, shares the same initial steps with $summa1$, up to the extraction of the following features: nearest-neighbors similarity features, Wikipedia prior features and features taken from the co-occurrences between all mention candidates. It then follows a different approach for the named disambiguation step. Following recent work towards jointly mapping words and entities into the same continuous vector space, we used the English distributed representations provided by Yamada et al. (2017) for English and trained Spanish representations using the same method. However, our ranking neural network architecture differs from Yamada et al. (2017). The differences will be outlined further in the disambiguation section. The nominal resolution and post-processing step is the same as in $summa1$.

Algorithm 1 $summa1$ Entity Linking System

```
High-precision sub-string match mention coreference
Candidate generation
Candidate rank step #0: information retrieval engine (kNN algorithm) + prior statistics
Candidate re-rank step #1: accounting for co-occurrences between all mentions candidates
Candidate re-rank step #2: accounting for coherence
Attempt to fix NIL mentions with edit distance threshold
Nominal mentions disambiguation
Global NIL clustering and cross-document coherence
```

Algorithm 2 $summa4$ Entity Linking System

```
High-precision sub-string match mention coreference
Candidate generation
Features generation: information retrieval engine (KNN algorithm) + prior statistics + mentions candidates co-occurrences
Distributed representation neural network disambiguation
Nominal mentions disambiguation
Global NIL clustering and cross-document coherence
```

Run #5, $summa5$, only shares the initial high-precision sub-string match mention coreference step (mention clustering) and the final steps. Its candidate generation differs from $summa4$ in the way the candidate list is expanded to reach optimal coverage. The disambiguation step is similar to the one used in $summa4$, with the only difference, that no search engine features are used but a document level coreference feature and a feature describing the level of fuzziness that was used to retrieve a

[4] https://github.com/Priberam/TurboParser

candidate. As in $summa4$, the nominal resolution and post-processing step is the same as in $summa1$. $Summa5$ can be described in the Algorithm 3.

Algorithm 3 $summa5$ Entity Linking System

```
High-precision sub-string match mention coreference
Candidate generation (different from run #1 and #4)
Distributed representation neural network disambiguation
Nominal mentions disambiguation
Global NIL clustering and cross-document coherence
```

Entity linking indexes In the search-engine-assisted runs, due to the necessity of linking recognized mentions to known entities, such task is intertwined with an auxiliary procedure, which involves building a database, to be queried in run-time, in which all linkable entities must be stored. For every language, we generated two information-retrieval indexes using Wikipedia as the source of information. The first index stores the content of each entity Wikipedia page as a bag of words, lemmas and detected entities in an inverted index. A second index is created, using the anchors information. Wikipedia anchors are used to discover alternative names to entities, to extract conditional probabilities of entities given those names and to derive co-occurrence models of mention/entity pairs. Accordingly, each record from such index corresponds to a unique mention surface, and stores the information of all the named entities that were linked from its anchor's occurrences, as for example, $p(e|m)$, i.e., probability of an entity given a mention m and the list of co-occurrent entities with each pair mention/entity.

Mention coreference For each mention, we perform a high-precision coreference step at the document level by linking all the surface mentions which are substrings of other mentions' forms. To preserve agreement within the coreference clusters, we heuristically reassign some entity types with a voting strategy.

Candidate generation Two candidate generation mechanisms were implemented: one for $summa1$ and $summa4$ and another for $summa5$. In $summa1$ and $summa4$, for each mention, the candidates are generated using the less ambiguous mention (defined as the one with the largest span) in the corresponding coreference cluster. Then, the candidate generation is performed based on the anchors' statistics in Wikipedia. In addition, for mentions with fewer candidates (less than 50), we also consider as candidates the entities whose titles have all the words of the query mention. If even after such procedure, the number of candidates is less than 10, the search for candidate entities is performed in an alternate mention index, in case of existence (i.e., another index from other language that shares multiple entity surface forms and/or the same alphabet).

In $summa5$, we only query the anchor index for the candidate generation. First, we expand the candidate list by querying the index with increasingly fuzzier searches on the anchor strings. The candidate set is then augmented with a second round of queries, where we restrict the set of possible candidates to the highest ranked candidates from all mentions from the current document. The score for ranking the candidates is computed from the normalized TF-IDF, boosted by the $log(\#(e|m))$.

The criterion for expanding the candidate list with increasingly fuzzier searches is the minimum number of total inlinks, i.e. $\#(*|m)$, for the set of candidates for a mention. This hyperparameter was tuned on training data to reach maximal coverage and was set to 400. The fuzziness level of a query are: full string match with an anchor, partial token overlap, partial token overlap with edit distance 1 and prefix length 2.

Candidate ranking Let $c_{i,k}$ be the k^{th} candidate of mention m_i and $s_{search}(c_{i,k}, m_i)$ be the score of a nearest-neighbors search engine procedure that reflects the similarity of the mention's document with the text of candidate $c_{i,k}$ composed by its Wikipedia title and body. In the 3^{rd} step of Algorithm 1, the candidates of each mention m_i are initially sorted according to this ranking score $s_{search}(c_{i,k}, m_i)$.

A model is applied, and a score expressed by

$$s_{model1}(c_{i,k}, m_i) = \sum_{j=1}^{n} \theta_j * c_{i,k_j}$$

is obtained, in which c_{i,k_j} are $c_{i,k}$'s features generated by the polynomial expansion of the nearest-neighbors similarity features and prior features, such as probability of an entity given a mention, and θ_i are the feature weights, trained with a pairwise approach, using SVM-rank (Joachims, 2002). For such training, a file with a list of training queries is generated (one query per mention), in which each line features a mention candidate, with its features and a target value, 1 if that candidate matches the gold one, and 0 otherwise.

Afterwards, a co-occurrence feature is computed for the top-ranked candidates for all mentions. Mention candidates co-occurrence feature values will be greater whenever those entities co-occur with more mentions and smaller the more ambiguous those mentions are (mentions with a higher number of candidates).

In the 4^{th} step, another model is applied and a new score, $s_{model2}(c_{i,k})$, with the same expression as the one in the previous step is obtained, but in this case with the additional co-occurrence feature. After the reorder by the score s_{model2}, one last reorder step is applied, accounting for coherence, as discussed in the next section.

Coherence re-rank Contrary to other state-of-the-art entity linking methods that favor solutions in which the entities of a same document are related with each other and that consider all possible combination of mentions candidates (being therefore, NP hard, (Kulkarni et al., 2009)), prior work typically relax the general collective formulation either by using continuous formulations (Kulkarni et al., 2009) or by identifying sets of mentions or entities that are somehow involved in a semantic relation (Hoffart et al., 2011; Ratinov et al., 2011; Sil et al., 2015; Pan et al., 2015) to tackle this problem of complexity. In this step we focus on the top 10 candidates obtained from the previous step and re-rank them to favor coherence. Our envisaged coherence model resolves each mention independently. To achieve coherence, the score of a mention's candidate is influenced by its coherence with all the candidates of the other mentions in the text:

$$s_{coherence}(c_{i,k}, m_i) = \sum_{j \neq i,l} \frac{s_c(c_{i,k}, c_{j,l})}{|C_j|}, \quad (1)$$

where C_j is candidate list of mention m_j and $s_c(c_{i,k}, c_{j,l})$ is a score that accounts for the coherence between the candidate under evaluation ($c_{i,k}$) and the l^{th} candidate of other mention m_j ($c_{j,l}$), and which is given by:

$$s_c(c_{i,k}, c_{j,l}) = \begin{cases} 1 + \frac{k}{p_{j,l}}, & c_{i,k}, c_{j,l} \text{ share a link} \\ \frac{1}{2} + \frac{k}{p_{j,l}} & \text{otherwise,} \end{cases} \quad (2)$$

where $p_{j,l}$ is the position of candidate $c_{j,l}$ according to the previous ranking score and k is a constant that represents the number of candidates considered for coherence. This coherence score was empirically designed to consider both coherence (as the existence or absence of a link) and information regarding previous candidate order.

Our coherence model, in Equation 1, is similar with the model that was independently proposed by Globerson et al. (2016).

Distributed representations neural network disambiguation Our implemented architecture uses a simple multi-layer perceptron (MLP) classifier with the learned representations, both from the text and from candidate entities, as features, composed by a relu hidden layer with dropout, followed by a softmax output layer. Contrary to Yamada et al. (2017), we also add context features and create an intermediate representation of the external features with a hidden layer. Figure 2 portraits a graphical representation of how these features are concatenated to form the candidate representation.

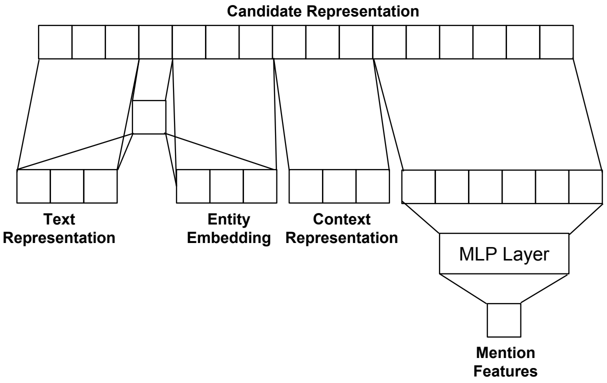

Figure 2: Candidate representation for the neural network disambiguation

For $summa4$ and $summa5$ the network is fed with

- the text representations,

- the entity representations,
- the dot product between the text representations and the entity representations,
- the context representations.

For $summa4$ we add the following intermediate representations of the features from the previous step:

- NN-search-engine features,
- prior features,
- co-occurrence features.

For $summa5$ we add the following features:

- Indicator, if this candidate was ranked first prior to the disambiguation for any other mention from this document,
- the level of fuzziness that was used to generate this candidate,
- prior features,
- character bigram overlap between the mention and the prefix and the suffix of the anchor.

NIL disambiguation with edit distance threshold After the named entity disambiguation, for every mention that was not linked to any KBID, we generate a new set of candidates for such mentions, where their surface is compared with the other disambiguated named mention surfaces, their entities titles and those entities links' titles in the KB. If the Levenshtein distance between the unlinked mention surface and those strings is smaller than 2, the corresponding entities get through to the next phase, in which these mentions are disambiguated in the exact same steps described in Algorithm 1, up to the co-occurrences re-rank step (steps 3 and 4). During this procedure, previous disambiguated mentions are locked and their selected KBIDs are not allowed to change.

Nominal mentions disambiguation The nominal resolution is a step performed after the named mention resolution. Two baseline approaches were developed for nominal entity linking: the first one, applied in $summa4$ and $summa5$, with the rationale that a nominal mention is followed by a previous mentioned named entity, aims at connecting a nominal mention to the left-closest disambiguated named mention entity that shares the same NER tag.

The other baseline approach, implemented in $summa1$ follows the same procedure used in the named resolution (and in the named NIL disambiguation), but restricts the set of possible candidates to the set of disambiguated named mentions entities and their links' entities (the procedure is the same described in the previous section).

Global NIL clustering and cross-document coherence Finally, the last step builds on top of the last coherence step to promote a new type of coherence that works at a corpora level. The underlying idea of this step is to promote coherence along the entities that co-occurred (with the same mention surface + candidate pair) in different documents.

Let, for each mention m_i, $D(m_i)$ be the set of the entities to which the other mentions ($m_{j \neq i}$) in the document link to. For each entity $e_{i,k}$ to which the surface of mention m_i links to in the full corpus, let $C(e_{i,k}, m_i)$ be the set of entities that co-occur in documents where the surface form of m_i connects to e_{ik}. We define the cross-document coherence score as

$$s_{cdc}(e_{i,k}, m_i) = J(D(m_i), C(e_{i,k}, m_i)), \qquad (3)$$

where $J(.)$ is the Jaccard similarity:

$$J(A, B) = \frac{A \cap B}{A \cup B}.$$

Each mention m_i is finally linked to the entity $e_{i,k*}$ with the highest cross-document coherence score, in Equation 3.

3 Results

3.1 Entity Recognition evaluation

We evaluated the impact of using available datasets for NER in TAC 2016 test set. The results reported in Table 1 show that the TAC train data alone could not achieve a great performance at NER and NERC F-scores. The usage of TAC Ontonotes and ACE improves the results as they add more examples to the training step.

We believe the access to more datasets is a crucial factor to improve the NER and NERC scores on TAC. The best NER system from TAC 2016, USTC team (Liu et al., 2016), reported they had access to an in-house built dataset which led to their top performance.

Table 1: Impact of the dataset for English named entity recognition in TAC 2016

Dataset	NER	NERC
TAC	83.37	79.04
+ Ontonotes	**87.35**	83.82
+ ACE	87.26	**83.97**

For Spanish, the same was verified. Table 2 compares two approaches: one using a simple LSTM+CRF model and another using our joint ensemble approach reported in Section 2.2.2. Our joint ensemble approach allowed us to use the English dataset which improved the NER and NERC F-scores.

Table 2: Impact of the approach and dataset for Spanish named entity recognition in TAC 2016

Approach	Dataset	NER	NERC
Simple	TAC Spanish	82.92	79.51
Joint ensemble	TAC Spanish + TAC English + Ontonotes	**85.58**	**82.69**

3.2 Entity Linking evaluation

Step ablation Regarding the Nearest-Neighbors-assisted ruled-based Entity Linking system, submitted as $summa1$, we present the contribution of each step to its performance in Table 3. Relevant metrics are reported with the ENG and NAM filters, at the end of steps 3, 4, 5 and 8. Every step led to an improvement across all metrics, which offers a cumulative noticeable improvement.

As it can be seen, the best performance is achieved in the final step, with a cumulative improvement across all steps of 2.4% with regard to mention detection, type classification and linking (NERLC), 2.0% regarding document entity tagging (KBIDs) and 2.9% with regard to clustering metrics: detection and clustering (CEAFm) and detection, clustering and type classification (CEAFmC). The biggest improvements were due to 1) the SVM re-rank with the co-occurrence features, contributing with 1% F1 improvement to the entity linking metrics, as the co-occurrences prove to be relevant for the disambiguation; and 2) the NIL clustering and corpora-level coherence, contributing with around 1.7% F1 improvement to the clustering metrics, as in this step NIL mentions are clustered together and some mentions are re-linked to other entities that have higher cross-document coherence scores, which leads to better clusters.

Feature ablation Contrary to Yamada et al. (2017), in the system represented by $summa4$, we added context representations and its contribution to the performance is presented in Table 4. Such results prove that the context features clearly contribute positively for the linking and clustering performance.

Table 5 evaluates the contribution of each set of external features to the $summa4$ system performance in the TAC KBP ELD 2016 evaluation corpus. It is worth noticing that the difference between training and testing with no external features and with all features is less or equal to 0.6% in almost all metrics (with the exception of the KBIDs metric), which points to the fact that the distributed representations of the text, entities and context already provide much of the knowledge necessary for the disambiguation. The best result is obtained with just prior and co-occurrence features, as opposed to using prior, co-occurrence and also the nearest-neighbors similarity features, which points to the fact that from the set of external features, they are the most relevant ones and that when training with all those external features, the system over-fitted to the training data.

Nominal resolution Our contribution to the nominal resolution is small and the results are not up to par with the ones in the named resolution. However, experiments with nominal resolution show us that with the nearest-neighbors search engine system we are able to attain circa 0.92 recall F1 score, using the set composed by the top-3 ranked candidates retrieved for all the mentions in the document. It is yet to be found a way to leverage such high recall to improve precision without reducing too much the recall, as it happens in both our baseline approaches, where we attained around 0.25 both in precision and recall, with the ENG-NOM filters. Future work may involve experimenting with distributed representations of text and entities for nominal resolution.

Official scores For TAC KBP EDL 2017 we submitted three runs, as previously described. Regarding Monolingual English EDL (Table 6), our

Table 3: Step ablation in in Nearest-Neighbors-assisted ruled-based EL pipeline system (*summa*1's system), for the TAC KBP EDL 2016 evaluation corpus, with ENG-NAM filter.

	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
first svm-rank with nearest-neighbors similarity features (step-3)	0.887	0.855	0.771	0.768	0.822	0.795
second svm-rank with nearest-neighbors similarity and co-occurrence features (step-4)	0.887	0.855	0.780	0.784	0.831	0.804
document-level coherence (step-5)	0.887	0.855	0.783	0.788	0.834	0.807
NIL clustering and corpora-level coherence (step-8)	0.887	**0.856**	**0.795**	**0.798**	**0.851**	**0.824**

Table 4: Impact of context representations in MLP classifier disambiguation (*summa*4's system), for the TAC KBP EDL 2016 evaluation corpus, with ENG-NAM filter, using no external usage of features.

	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
without context representations	0.887	0.856	0.774	0.804	0.829	0.802
with context representations	0.887	0.856	**0.788**	**0.817**	**0.842**	**0.815**

Table 5: External feature ablation in MLP classifier disambiguation (*summa*4's system), for the TAC KBP EDL 2016 evaluation corpus, with ENG-NAM filter.

	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
no external features	0.887	0.856	0.788	0.817	0.842	0.815
just prior and co-occurrence features	0.887	0.856	**0.795**	**0.839**	**0.852**	**0.825**
prior, nearest-neighbors similarity and co-occurrence features	0.887	0.856	0.794	0.838	0.848	0.820

NERC placed us in the 6^{th} place, with 0.784 F1 score. We then achieved 0.653 (5^{th} place) for the entity linking NERLC metric and 0.674 (4^{th} place) for clustering CEAFmC metric. Spanish results (Table 8) are similar: 0.750 F1 score for NERC (6^{th} place), 0.594 F1 score for NERLC (also 6^{th} place) and 0.619 F1 score CEAFmC metric (5^{th} place).

More impressive results were attained with the NAM filter, specifically for Monolingual English EDL (NAM). Although we start from a 4^{th} place in the NERC metric with 0.861 F1 score (best score was 0.873), we were able to achieve the 1^{st} place both in entity linking NERLC metric and clustering CEAFmC metric, with 0.794 and 0.831 F1 scores, respectively (Table 7). In the Monolingual Spanish EDL (NAM), we rank 8^{th} place in the NERC metric with 0.816 F1 score (best score was 0.873) and achieved 3^{st} place both in entity linking NERLC metric and clustering CEAFmC metric, with 0.745 and 0.788 F1 scores, respectively (Table 9).

4 Conclusions

This paper described the contribution of SUMMA to the NIST TAC-KBP 2017. In this second year, we competed in the EDL track. We developed work on top of our TAC KBP submission of last year, where our main contributions to the track were our co-occurrence model and coherence steps, both intra and inter-document, in which the latter coherence score favors agreement between bags-of-entities along a corpus in an original approach.

We also submitted a language independent system to the EDL track, and although we still have a gap to close in NER and NERC F1-scores comparing with the best teams, we successfully managed to obtain the 1^{st} position in English Named disambiguation

Table 6: English results on the TAC-KBP EDL 2017 test data

Run	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
summa1	0.819	0.782	0.647	0.737	0.692	0.670
summa4	0.819	0.782	**0.653**	**0.748**	**0.695**	**0.674**
summa5	**0.820**	**0.784**	0.597	0.720	0.676	0.655

Table 7: English results with NAM filter on the TAC-KBP EDL 2017 test data

Run	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
summa1	0.885	0.859	0.784	0.773	0.849	0.827
summa4	0.885	0.859	**0.794**	**0.791**	**0.853**	**0.831**
summa5	**0.887**	**0.861**	0.716	0.757	0.831	0.809

Table 8: Spanish results on the TAC-KBP EDL 2017 test data

Run	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
summa1 *summa5*	0.801	0.750	0.590	0.696	**0.646**	**0.619**
summa4	0.801	0.750	**0.594**	**0.709**	0.644	0.618

Table 9: Spanish results with NAM filter on the TAC-KBP EDL 2017 test data

Run	NER	NERC	NERLC	KBIDs	CEAFm	CEAFmC
summa1 *summa5*	0.857	0.816	0.736	0.737	0.819	0.784
summa4	0.857	0.816	**0.745**	**0.755**	**0.821**	**0.788**

both in NERLC and CEAFmC evaluation metrics and 3^{rd} position in Spanish Named disambiguation, also in NERLC and CEAFmC metrics.

Acknowledgments

This work was partially supported by the SUMMA project – this project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688139, by Fundação para a Ciência e Tecnologia (FCT) through contract UID/EEA/50008/2013, through the LearnBig project (PTDC/EEI-SII/7092/2014), and the GoLocal project (grant CMUPERI/TIC/0046/2014).

References

Carlos Amaral, Adán Cassan, Helena Figueira, André Martins, Afonso Mendes, Pedro Mendes, José Pina, and Cláudia Pinto. 2008. Priberam's question answering system in qa@ clef 2008. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 337–344. Springer.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. *arXiv preprint arXiv:1705.02494*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, U.K., 27–29 July 2011, pages 782–792.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and

Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466.

Dan Liu, Wei Lin, Shiliang Zhang, Si Wei, and Hui Jiang. 2016. Neural networks models for entity discovery and linking. *CoRR*, abs/1611.03558.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Peteris Paikens, Guntis Barzdins, Afonso Mendes, Daniel Ferreira, Samuel Broscheit, Mariana S. C. Almeida, Sebastião Miranda, David Nogueira, Pedro Balage, and André F. T. Martins. 2017. Summa at tac knowledge base population task 2016. In *Proceedings of the Text Analysis Conference -TAC*, pages 1–9, Gaithersburg, Maryland USA.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Avirup Sil, Georgiana Dinu, and Radu Florian. 2015. The ibm systems for trilingual entity discovery and linking at tac 2015. In *Proc. Text Analysis Conference (TAC2015)*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning Distributed Representations of Texts and Entities from Knowledge Base. *ArXiv e-prints*, May.