SUMMA at TAC Knowledge Base Population Task 2016

Peteris Paikens* **Guntis Barzdins*** **Afonso Mendes†** **Daniel Ferreira†**
Samuel Broscheit# **Mariana S. C. Almeida†** **Sebastião Miranda†** **David Nogueira†**
Pedro Balage† **André F. T. Martins#,†**
*University of Latvia / News agency LETA
†Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
#Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
{amm,mla,summa}@priberam.pt, Peteris.Paikens@leta.lv

1 Introduction

Our submission to the NIST TAC-KBP-2016[1] is an initial attempt to apply our ongoing research on text analysis within SUMMA project[2] to TAC shared tasks. The goal of SUMMA is to develop a scalable and extensible media monitoring platform with an automatic knowledge base construction and cross-lingual capabilities, thus having a significant overlap with TAC-KBP tasks. For this first TAC participation, our system was only run for the Entity Discovery and Linking (EDL) and Cold Start Knowledge Base Population (KBP) tasks as a way to evaluate our initial system. In the next edition of TAC-KBP, we expect to participate with a more mature system.

The paper is organized as follows. Section 2 and Section 3 describe our contribution to the EDL and to the Cold Start KBP tracks, respectively. Experimental results are reported in Section 4, and Section 5 concludes the paper.

2 Entity Discovery and Linking

2.1 Submissions

Two systems were submitted to the first evaluation window of the EDL track. The first system, *summa1*, is an initial implementation of a language independent approach. The system is based on an implementation of SVM-Rank (Herbrich et al., 2000) trained with "universal" features, namely features obtained from pre-trained cross-lingual representations (Ferreira et al., 2016). Despite having a cross-lingual framework, due to evaluation window time constraints we submitted our results only for English. The second submission, *summa2*, is a ruled-based system for English, that evaluates the impact of several steps into the linking quality.

Since *summa2* outperformed *summa1* in the first evaluation window, in the second evaluation window we focused on an improved version of *summa2*, by adding a candidate ranking step based on nearest-neighbours retrieval and a novel cross-document coherence step. Ahead, this section provides a description of our final submission – *summa3*.

2.2 Entity Recognition and Labeling

Model and features. For detecting and labeling mentions, we use the named entity recognizer (NER) available within TurboParser[3] (Martins et al., 2013). This NER implements a linear sequential model whose features are based on the Illinois Entity Tagger (Ratinov and Roth, 2009).

Training data. As training set, we use the whole TAC-KBP 2015 training data and roughly one third of the Ontonotes. We use the Ontonotes' entity types corresponding to the TAC data (PER, ORG, FAC, LOC and GPE) plus the NORP type. Later, at the end of the linking phase, NORP mentions are assigned a TAC entity type, by mapping the DBpedia info of the selected entity to the five types of the task or, for NIL mentions, by setting the entity type to GPE.

We only focus on named entity mentions (NAM)

[1] https://tac.nist.gov/2016/KBP/
[2] http://www.summa-project.eu/
[3] http://www.cs.cmu.edu/~ark/TurboParser/

mentions, therefore we did not develop a strategy for detecting nominal (NOM) mentions.

Post-processing. As a post-processing step, we force to detect mentions that are marked in the text as being the authors of the articles, and we tag them with the PRE type.

We also apply a string matching procedure to capture mentions that were not recognized by the sequential model. In particular, we extract mentions with the exact same surface form 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, at the end of the linking stage, some of the entity types are also reassigned in order to promote label agreement after both the co-reference and the linking steps (see details in section 2.3).

2.3 Linking System

The mentions detected in Section 2.2 are linked to database entries according to the strategy described in Algorithm 1.

Algorithm 1 Linking System

1: Simple string match coreference
2: Candidate generation
3: Candidate rank: NN-search + prior statistics
4: Re-rank (top 8 candidates) accounting for coherence
5: NIL detection
6: Cross-document coherence

Coreference. First, we perform a high-precision coreference step at the document level, by linking all the mentions whose surface forms are substrings of other mentions' forms. For preserving the agreement within the coreference clusters, some entity types are then heuristically reassigned with a voting strategy.

Candidates generation. 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 candidates generation itself is performed based on the probability of an entity given a mention, $p_{\text{wp\_wl\_conll\_TAC}}(e|m)$, computed from statistics of the anchors in the following datasets: Wikipedia, Wikilinks[4], AIDA-CoNLL2003[5] and TAC-KBP training data. In addition to that, and for mentions with fewer candidates (less than 60), we also consider as candidates the entities whose titles have all the words of the query mention.

Candidates rank. Step-3 of Algorithm 1 starts by ranking the candidates using a nearest-neighbours (NN) search criterion. To this end, a query feature vector q_i is built for each mention m_i from the body of the source document, considering lemmas, heads and root words. Then, a similarity search operation is executed on a search index with the Wikipedia entities indexed with their corresponding Wikipedia body and Wikilinks text. Considering c_{ik} as the k^{th} nearest-neighbour candidate of mention m_i, this operation approximately computes $s_{sim}(c_{ik}, m_i)$, which is the similarity between q_i and c_{ik} in the feature space, using a ranking function based on Okapi BM25.

Based on the search similarity $s_{sim}(c_{ik}, m_i)$, we compute a preliminary ranking score

$$s_0(c_{ik}, m_i) = s_{sim}(c_{ik}, m_i) \cdot (1 + s_{\text{wp\_wl\_conll}}(c_{ik})), \tag{1}$$

where $s_{\text{wp\_wl\_conll}}(c_{ik})$ is a score related with the likelihood of candidate c_{ik}, computed as follows:

$$\begin{aligned} s_{\text{wp\_wl\_conll}}(c_{ik}) = & k_1 \cdot \log p_{\text{Wikipedia}}(c_{ik}) \\ & + k_2 \cdot \log p_{\text{Wikilinks}}(c_{ik}) \\ & + k_3 \cdot \log p_{\text{CoNLL-03}}(c_{ik}), \end{aligned}$$

where $p_{\text{Wikipedia}}$, $p_{\text{Wikilinks}}$ and $p_{\text{CoNLL-03}}$ are the probabilities of candidate c_{ik} extracted from the statistics in Wikipedia, Wikilinks and AIDA-CoNLL2003 corpora, respectively; and k_1, k_2 and k_3 are tunable positive parameters.

Finally, based on score s_0, step-3 of Algorithm 1 sorts the candidates of each mention using the following ranking score:

[4] http://www.iesl.cs.umass.edu/data/wiki-links

[5] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/

$$
\begin{aligned}
s_1(c_{ik}, m_i) = & s_0(c_{ik}, m_i) \\
& + k_4 \cdot p_{\text{TAC}}(c_{ik}|m_i) \\
& + k_5 \cdot p_{\text{TAC}}(c_{ik}|m_i) \cdot s_0(c_{ik}, m_i) \\
& + k_6 \cdot p_{\text{wp\_wl\_conll}}(c_{ik}|m_i) \\
& + k_7 \cdot p_{\text{wp\_wl\_conll}}(c_{ik}|m_i) \cdot s_0(c_{ik}, m_i),
\end{aligned}
$$

where c_{ik} is the k^{th} candidate of mention m_i; k_4, k_5, k_6 and k_7 are real valued positive parameters; $p_{\text{TAC}}(c_{ik}|m_i)$ is the conditional probability of candidate c_{ik} given its mention m_i, computed from the statistics in the TAC-KBP training dataset; and $p_{\text{wp\_wl\_conll}}(c_{ik}|m_i)$ is the same conditional probability computed from the concatenation of Wikipedia, Wikilinks and AIDA-CoNLL2003 corpora.

Re-rank for coherence State-of-the-art methods for entity linking use coherence models that favor solutions in which the entities within a same document are related with each other. The inference of a fully collective model, however, is NP hard (Kulkarni et al., 2009), since one must consider all possible combination of mentions candidates. To tackle this problem of complexity, 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).

In step-4 of Algorithm 1, we focus on the top 8 candidates obtained in step-3 and re-rank them to favor coherence. In contrast to previous work, our 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:

$$s_2(c_{ik}, m_i) = s_1(c_{ik}, m_i) \cdot \Big(1 + \sum_{j \neq i, l} s_c(c_{ik}, c_{jl})\Big), \tag{2}$$

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

$$s_c(c_{i,k}, c_{j,l}) = \begin{cases} \frac{1}{p_{ij}}, & c_{ik}, c_{jl} \text{ share a link} \\ \frac{1}{2p_{ij}}, & \text{otherwise,} \end{cases} \tag{3}$$

where p_{ij} is the position of candidate c_{jl} according to the previous ranking score $s_1(c_{jl}, m_j)$. 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 (2), is similar to the model that was independently proposed by Globerson et al. (2016).

NIL detection For documents with at least 10 mentions, we accomplish NIL detection by verifying that a mention has no coherence (measured as the existence or absence of links) with any of the other mentions in the text. After that, some of the NILs are linked to database entries, depending on the links of other mentions in the same coreference cluster. This NIL detection is latter improved in the cross-document coherence step.

Cross-document coherence Finally, step-6 of Algorithm 1 builds on top of step-5 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+candidate pair) in different documents.

Let, for each mention m_i, $\mathcal{D}(m_i)$ be the set of the entities to which the other mentions in the document ($m_{j \neq i}$) link to (according to step-5). For each entity e_{ik} to which the surface of mention m_i links to in the full corpus, let $\mathcal{C}(e_{ik}, m_i)$ 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

$$s_3(e_{ik}, m_i) = J(\mathcal{D}(m_i), \mathcal{C}(e_{ik}, m_i)), \tag{4}$$

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

$$J(\mathcal{A}, \mathcal{B}) = \frac{\mathcal{A} \cup \mathcal{B}}{\mathcal{A} \cap \mathcal{B}}. \tag{5}$$

Each mention m_i is finally linked to the entity, e_{ik*}, with the highest cross-document coherence score, in (4).

At the end of the linking system, we map the DBpedia labels of the selected entities to the five NER types of the task, and use them to reassign the types of the corresponding NORP mentions.

2.4 Future Directions

Our EDL system consists of several steps that were successfully engineered for the task, and whose parameters can be hand tuned. In the future, we expect to include machine learning in the EDL system. This would allow us to automatically learn the best configuration of parameters and to be able to easily use and test more features.

In a complementary line of research, we plan to use and develop new language-independent features in order to reach a final system which, in line with our *summa1* submission, would be suitable to process documents in different languages.

We also plan to improve our NER module, which appears to be an important bottleneck of the final EDL system.

3 Cold Start KBP

3.1 Motivation and system structure

Our motivation for the Cold Start KBP task was to test a hypothesis that the slot filling problem can be solved by general purpose semantic parsers without specific training data or parser customization. Due to our earlier experience with semantic parsing with the Abstract Meaning Representation (Banarescu et al., 2013) formalism, we apply the top performing AMR parser that we developed for Semeval-2016 competition (Barzdins and Gosko, 2016) and attempt to map its output to the slots specified in Cold Start KB construction task as described in Algorithm 2.

Algorithm 2 KBP slot filling

1: Preprocessing and sentence extraction
2: AMR parsing with a CAMR parser
3: Entity Detection and Linking system
4: Mapping the AMR concept instances to the EDL entities
5: Mapping the AMR predicates to appropriate slot fillers

3.2 Submissions

We submitted two runs obtained by an identical process but differing only in the set of EDL data used. The *summa_KB_ENG_1* run was obtained by using the EDL data from submission *IBM1*, which we believed to be a state of art EDL result from other competitors; and we built the *summa_KB_ENG_2* run from our own team *summa2* EDL submission described in section 2.1.

3.3 AMR parsing

AMR parsing is done by a customized version of the CAMR parser (Wang et al., 2015b; Wang et al., 2015a) as used in our earlier experiments on the Semeval challenges. No extra training data or additional tuning of the AMR layer were used for this task, as the goal was to evaluate potential applications of general purpose semantic parsing. AMR parsing was performed on a sentence-by-sentence level, with no intra-document coreference resolution. It was expected that integrating the EDL system results would link these references but this did not materialize (especially for nominal mentions and pronouns), resulting in significantly lowered accuracy. For future submissions this would be a key issue that needs a solution as rather frequently the required answer was not reached because this lack of intra-document linking of AMR nodes.

3.4 Entity mapping between AMR and EDL data

As the AMR annotation results in a very different set of entities than the EDL guidelines, the entity mapping is not trivial.

AMR entities The initial set of KBP entities is populated by the instances of AMR concept classes listed in Table 1. In most cases these entities are linked to a particular set of tokens, however that is not always true - often AMR identifies entities that have particular role in some predicate, but are not explicitly mentioned in that part of the sentence and would require a document level coreference resolution between AMR graphs of the document (like multi-sentence AMR construct).

Entity linking Linking of these entities with the appropriate entities identified by the EDL systems is currently done based on boundary overlap - an exact boundary match is not required. In many cases no appropriate entities are found, so we insert new entities that were detected in semantic parsing but were not present in EDL data. Entities from EDL data that could not be linked to appropriate nodes in the AMR graphs were not included in the KBP submission under assumption, that they are not relevant to the relations in this slot filling task; therefore the recall measure of entities in the official scoring is low and reflects only the entities identified by the AMR parser.

AMR concept class	TAC entity type
person	PER
country	GPE
state	GPE
province	GPE
city	GPE
town	GPE
organization	ORG
religious-group	ORG
company	ORG
government-organization	ORG

Table 1: AMR entity mapping

3.5 Predicate mapping between AMR and KBP slots

The actual slot filling is performed by scanning AMR data for a specific subset of AMR concepts that 'trigger' one of the targeted slot filling sets. For this set of concepts we developed a heuristic transformation that scans surrounding nodes of the semantic graph and maps the identified AMR links to particular types of knowledge base slots. The actual mappings are illustrated in Table 6.

As the AMR parser model is generic and not adapted to the particular needs of TAC KBP slot filling task, some slots have no corresponding concepts in AMR data and thus cannot be filled by this approach. However, the more popular types of data such as employment and relationships have a good match between these systems.

It should be noted that the resulting mappings generally are 1-to-n, as the AMR predicates are n-ary relations (similar to the annotation concept used in the Event Nugget track) and can imply multiple different binary relations because both relationship directions need to be considered separately (e.g. the symmetric relationships of employee and employer) and also because the KBP slot filling annotation marks otherwise identical slots differently depending on the entity type.

The transformation process needs to consider additional information from the whole predicate. For example, a employment relationship between a person and a company may result in filling either the slot org:employees_or_members or org:top_members_employees, and the distinction can be made by considering the position label of that AMR predicate. In a similar manner, the predicate for personal relationships has a field (ARG2 in the annotation) describing the type of relation, so it can be transformed to the appropriate choice of KBP slot.

Certain slots can be filled by considering relations that in the AMR parse graph are syntactic predicates as opposed to semantic ones - for example, the residence slots often are described by having a country or location entity as possessive modifier of the person or company.

4 Results

4.1 Entity Discovery and Linking

NER evaluation. This section evaluates the impact of the NER post-processing step into the quality of the CRF model, using the TAC-KBP 2015 test set. Table 2 shows two official metrics for NER[6], computed for the output of the base NER (using a CRF model), after the coreference step of Section 2.3, and at the end of the linking system.

The mentions that were bootstrapped in the post-processing of Section 2.2 lead to a considerable increase (5.4%) of the `NER` F_1 score. This improvement was mainly (but not only) due to a strong increase in the recall. `NERC` measure suffered a considerable boost due to a better detection of mentions and a suitable reassignment of types based on the coreference clusters. This measure is also improved at the end of the system, when the NORP tags are reassigned based on the linking results. Overall, we got a positive impact of

[6] see (Ji et al., 2015) for details.

more than 5% in both NER measures, validating our system design options.

	NER	NERC
CRF model	73.8%	67.9%
+expansion+coreference	79.2%	72.2%
+NORP reassignment	**79.2**$\%$	**73.9**$\%$

Table 2: Impact, of NER post-processing steps, in both NER and NERC F_1 scores, using EN and NAM filters.

Step ablation. To evaluate the impact of each step of Algorithm 1 in the final linking system, Table 3 reports various metrics (with EN and NAM filters) at the end of steps 3 4 5 and 6. Each system step leads to cumulative improvements in the global measure NERLC, which accounts for mentions detection, type classification and linking. We also verify that, for all the measures, the final stage of the algorithm is the one with the highest F_1 value. For this outcome, we point out the final stage of cross-document coherence, which has a consistent positive effect into all of the metrics.

Steps 4 and 5 do not always lead to improvements, by themselves. In spite of that, we have experimentally verified that these steps have a final positive impact, even when a local evaluation may indicate them to be disadvantageous. One situation where it is easy to understand this effect, is the decrease of the NENC F_1 score after NIL detection. This decrease is mainly due to an increase in the number of the NIL mentions (lowering the precision), some of which are further relinked to the correct entity when accounting for cross-document coherence.

Regarding coreference evaluation, measure CEAFm suffers a considerable improvement in step-5, when, after detecting NIL mentions, some of them are resolved to entities based on the co-reference clusters. Finally, our cross-document step is also useful for coreference resolution.

System evaluation. Table 4 evaluates our system performance on TAC2016 test data, using the EN-NAM filter. Regarding mention detection, whose quality is reflected in metrics NER and NERC, we only scored 8^th out of 11 teams. Despite of starting with this large disadvantage, our scores increase considerably (improving three positions in the classification rank) when we account for the linking quality (see metrics NERLC, KBIDs and CEAFm). This fact indicates that we have a high performing linking system. To validate this intuition, we run our linking step on top of the mentions detected by the USTC system (Liu et al., 2016). USTC team achieved the highest scores in most of the metrics of the shared task, including those regarding mention detection. From this comparison (whose results are in the last columns of Table 4), we conclude that our linking system is on a pair with the best systems in the competition.

4.2 Cold Start KBP

The official results of KBP evaluation are shown on Table 5, ranking at 13^{th} place out of 19 teams. The low recall rate is rather disappointing, however, the error analysis indicates that this is largely caused by faults in the linking process between AMR graph nodes and EDL entities as discussed in 3.4. On the other hand, the system achieves good precision, so with appropriate fixes it could be competitive in the next iteration of TAC KBP.

5 Conclusions

This paper described the contribution of SUMMA submissions to the NIST TAC-KBP 2016. In this first year, we competed in the EDL and cold start KBP tracks.

Regarding the EDL track, our main submission was a rule-based system, whose steps were empirically validated. As main contribution to the track, we point out our coherence step that treats each mention independently and the impact of an original corpora-level coherence score, which favours agreement between bags-of-entities along a corpus. We also attempted to submit a language independent system to the EDL track, but we did not have time for making a final competitive submission.

Regarding cold start KBP, we establish a proof of concept that the KBP slot filling task may be approached by using general purpose semantic parsing models. While current results indicate a number of technical challenges in transformations between these very different semantic models, this

	Basic Rank (step-3)	Intra-Doc. Coherence (step-4)	NILL Detect. (step-5)	Cross-Doc. Coherence (step-6)
`NERLC`	61.1%	62.3%	62.4%	**64.7**%
`NELC`	61.6%	61.2%	62.3%	**63.9**%
`NENC`	59.8%	65.1%	64.7%	**66.7**%
`KBIDs`	68.4%	68.1%	70.4%	**70.8**%
`CEAFm`	58.2%	57.3%	69.7%	**71.7**%

Table 3: Evaluation, on the TAC-KBP 2015 test data, of our system at the end of steps 3, 4, 5 and 6. Each row shows F_1 scores for an official measure (Ji et al., 2015), computed using EN and MAN filters.

	Summa3	rank	USTC (2016)	*Summa3* (USTC mentions)
`NER`	83.1%	$8^t h$	**90.6**%	**90.6**%
`NERC`	76.1%	$8^t h$	**87.8**%	**87.8**%
`NERLC`	66.4%	$5^t h$	79.2%	**79.8**%
`KBIDs`	70.8%	$6^t h$	**81.1**%	81.0%
`CEAFm`	74.4%	$5^t h$	83.2%	**83.3**%

Table 4: EDL evaluation on TAC-KBP 2016 test data, using EN-NAM filter. First column: our final system; second column: position of our system in the competition; third column: USTC system; last column: our linking system using USTC mentions.

	`Prec`	`Recall`	`F1`
LDC MAX 0 hop	45.0%	2.9%	5.5%
LDC MAX ALL	40.0%	2.2%	4.1%

Table 5: Accuracy of the resulting knowledge base SUMMA2 submission.

approach shows potential and we expect to provide a significantly improved implementation in the next issue of TAC KBP.

Acknowledgments

This work was mainly 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. This work was also supported by Fundação para a Ciência e Tecnologia (FCT) through contracts UID/EEA/50008/2013, GoLocal project (grant CMUPERI/TIC/0046/2014) and the Latvian state research programme SOPHIS (Project No. 2).

References

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of Linguistic Annotation Workshop*.

G. Barzdins and D. Gosko. 2016. Riga at semeval-2016 task 8: Impact of smatch extensions and character-level neural translation on amr parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*.

Daniel Ferreira, André Martins, and Mariana S. C. Almeida. 2016. Jointly learning to embed and predict with multiple languages. In *Annual Meeting of the Association for Computational Linguistics - ACL*, August.

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

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 2000. Large margin rank boundaries for ordinal regression.

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 Conference on Empirical Methods in Natural Language Processing – EMNLP*, pages 782–792.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Text Analysis Conference*.

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. The USTC NELSLIP Systems for Trilingual Entity Detection and Linking Tasks at TAC KBP 2016. In *Proc. Text Analysis Conference (TAC2016)*.

André F. T Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

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*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL*, pages 147–155.

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*, pages 1375–1384.

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)*.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based amr parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 857–862.

Chuan Wang, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015b. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

AMR concept	TAC KBP slot
have-org-role-91	per:title
have-org-role-91	per:employee_or_member_of
have-org-role-91	org:employees_or_members
have-org-role-91	gpe:employees_or_members
have-org-role-91	per:top_member_employee_of
have-org-role-91	org:top_members_employees
leader	per:top_member_employee_of
leader	org:top_members_employees
mod	per:countries_of_residence
mod	gpe:residents_of_country
mod	per:cities_of_residence
mod	gpe:residents_of_city
mod	per:statesorprovinces_of_residence
mod	gpe:residents_of_stateorprovince
mod	org:countries_of_headquarters
mod	gpe:headquarters_in_country
mod	org:cities_of_headquarters
mod	gpe:headquarters_in_city
mod	org:statesorprovinces_of_headquarters
mod	gpe:headquarters_in_stateorprovince
have-rel-role-91	per:spouse
have-rel-role-91	per:siblings
have-rel-role-91	per:children
have-rel-role-91	per:parents
have-rel-role-91	per:other_family
study-01	per:schools_attended
study-01	org:students
shareholder	per:holds_shares_in
shareholder	org:holds_shares_in
shareholder	org:shareholders
die-01	per:date_of_death
die-01	per:city_of_death
die-01	per:country_of_death
die-01	per:stateorprovince_of_death

Table 6: AMR predicate mapping