Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

# Joint Posterior Revision of NLP Annotations via Ontological Knowledge

**Marco Rospocher** and **Francesco Corcoglioniti**
Fondazione Bruno Kessler (FBK-irst)
{rospocher, corcoglio}@fbk.eu

## Abstract

Different well-established NLP tasks contribute to elicit the semantics of entities mentioned in natural language text, such as Named Entity Recognition and Classification (NERC) and Entity Linking (EL). However, combining the outcomes of these tasks may result in NLP annotations — such as a NERC organization linked by EL to a person — that are unlikely or contradictory when interpreted in the light of common world knowledge about the entities these annotations refer to. We thus propose a general probabilistic model that explicitly captures the relations between multiple NLP annotations for an entity mention, the ontological entity classes implied by those annotations, and the background ontological knowledge those classes may be consistent with. We use the model to estimate the posterior probability of NLP annotations given their confidences (prior probabilities) and the ontological knowledge, and consequently revise the best annotation choice performed by the NLP tools. In a concrete scenario with two state-of-the-art tools for NERC and EL, we experimentally show on three reference datasets that for these tasks, the joint annotation revision performed by the model consistently improves on the original results of the tools.

## 1 Introduction

Text Understanding and many approaches for Knowledge Extraction and Ontology Population (e.g., NewsReader [Vossen *et al.*, 2016], PIKES [Corcoglioniti *et al.*, 2016]) rely on well-established NLP tasks for eliciting the semantics of entities mentioned in natural language text. These tasks — such as Named Entity Recognition and Classification (NERC), Entity Linking (EL), and Semantic Role Labeling (SRL) — have been extensively investigated by the NLP community, and high-performance methods and tools specifically tailored to tackle each of these tasks have been proposed over the years.

However, despite the good performances on the task they are designed for, combining the outcome of these tools' analyses may result in unlikely or even contradictory information. Consider for instance the sentence "Mr. Washington was runner-up at Wimbledon in 1996". Here, the entity mention "Washington" refers to the tennis player Malivai Washington.[1] However, using two state-of-the-art NLP tools, one for NERC (Stanford NER[2]) and one for EL (DBpedia Spotlight[3]), the first correctly identifies "Washington" as a person, while the second wrongly links it to the DBpedia entity corresponding to "Washington (the US State)". As another example, on the sentence "The GW Bridge is a suspension bridge over the Hudson." the NERC tool wrongly identifies the mention "GW Bridge" as an organization while the EL one correctly links it to "George Washington Bridge".

The work presented in this paper contributes to the problem of assessing and improving the coherence of the annotations produced for various NLP tasks. The first contribution is a general probabilistic model — JPARK[4] — that, given an entity mention in a text, leverages some background ontological knowledge to capture:

1. the relation among various NLP mention annotations;
2. the ontological entity classes implied by the annotations;
3. the background ontological knowledge those classes may be consistent with.

In particular, the model allows estimating *a posteriori* the overall confidence of a certain combination of NLP annotations (one annotation for each tool) on a mention, given the background ontological knowledge considered. Such overall confidence is expressed in terms of (i) the *a priori* confidences of each annotation, provided by the NLP tools, and (ii) the probability of predicting, given an annotation, some ontological classes for the entity denoted by the mention, a quantity that can be learned from training data.

The second contribution is a concrete instantiation of the general probabilistic model for NERC and EL, using YAGO [Hoffart *et al.*, 2013] classes as background ontological knowledge. In particular, we show how to use a reference dataset for NERC and EL, namely AIDA CoNLL-YAGO [Hoffart *et al.*, 2011], to estimate the probability that, given a NERC or EL annotation for a mention, some ontological classes characterize the entity denoted by that mention.

[1] http://bit.ly/MaliVai (accessed on Apr 26, 2018)
[2] http://bit.ly/demoNER (accessed on Apr 26, 2018)
[3] http://bit.ly/db-spot (accessed on Apr 26, 2018)
[4] Joint Posterior Annotation Revision with Knowledge

As third and final contribution, we show how to operationally apply the instantiated model for NERC and EL in order to revise the annotations produced by two state-of-the-art tools for NERC (Stanford NER [Finkel *et al.*, 2005]) and EL (DBpedia Spotlight [Daiber *et al.*, 2013]). In details, given multiple NERC and EL candidate annotations (i.e., alternative annotations for each task, weighted with a confidence score by the corresponding tool) on the same entity mention, the model selects the ⟨NERC annotation, EL annotation⟩ combination that maximizes the aforementioned a posteriori overall confidence. By applying the model on three reference evaluation datasets for NERC and EL, we experimentally show that the posterior revision performed by the model consistently improves on the original results of the tools.

While some other approaches have investigated joint analyses of multiple NLP tasks in order to improve the performances on each of them, mainly training joint models for NERC and EL (e.g., [Stern *et al.*, 2012; Leaman and Lu, 2016; Nguyen *et al.*, 2016]), to the best of our knowledge this is the first posterior probability, ontological powered approach aiming to assess and improve the coherence of the annotations separately produced for various NLP tasks.

The paper is structured as follows. Section 2 presents the general probabilistic model. Section 3 describes how to build and train the model for NERC and EL. Section 4 reports the empirical assessment of using JPARK to improve the performance of Stanford NER and DBpedia Spotlight. Section 5 discusses relevant related works, while Section 6 concludes.

## 2 General Approach

In JPARK, we are interested in the probabilistic relations among five variables defined as follows:

- $m$ is an entity mention, a complex object whose internal structure, relevant in NLP tasks, is here ignored;
- $\boldsymbol{a} = (a_1 \dots a_n)$, $a_i \in A_i$ is a vector of NLP annotations for the mention, where $n$ is the number of different NLP tasks considered (e.g., $i = 1$ for NERC, $i = 2$ for EL) and $A_i$ is the set of all possible alternative annotations for the $i$-th task (e.g., $A_1 = \{\text{PER, ORG, LOC, MISC}\}$);
- $B$ is the background knowledge (i.e., knowledge not embedded in $m$) considered by NLP tools in their annotations, such as gazetteers and various training material;
- $K$ is the ontological knowledge here considered, relevant specifically for the joint execution of tasks, consisting of class information and popularity for entities;
- $C$ is the set of ontological classes associated to the entity denoted by the mention, consistently with $K$.

The confidence scores resulting from NLP tasks can be interpreted as — or calibrated to [Zadrozny and Elkan, 2002] — the probabilities $\mathrm{P}(a_i|m, B)$, for all tasks $i$ and annotations $a_i \in A_i$ (e.g., $P(\text{PER}|m, B) = 0.7$, $P(\text{ORG}|m, B) = 0.2$, …, for $i =$ NERC, values from tool output). In JPARK we want to account also for the ontological knowledge $K$, proposing a discriminative model for $\mathrm{P}(\boldsymbol{a}, C|m, B, K)$ that enables estimating posterior annotation confidences, and from that the optimal annotations and/or ontological classes.

To devise JPARK, we start postulating three approximate but necessary conditional independence assumptions:

($\text{ci}_1$) $a_1, \dots, a_n$ are conditionally independent given $m$, $B$, $K$, $C$, i.e., $\mathrm{P}(\boldsymbol{a}|m, B, K, C) = \prod_i \mathrm{P}(a_i|m, B, K, C)$;

($\text{ci}_2$) $a_i$ and $K$ are conditionally independent given $m$ and $B$, i.e., $\mathrm{P}(a_i|m, B, K) = \mathrm{P}(a_i|m, B)$;

($\text{ci}_3$) $C$ and $\langle m, B\rangle$ are conditionally independent given $K$ and $a_i$, for all $a_i$, i.e., $\mathrm{P}(C|a_i, m, B, K) = \mathrm{P}(C|a_i,K)$.

Assumption ($\text{ci}_1$) captures the intuition that correlations among NLP tasks (e.g., PER type for NERC mostly occurring with EL person entities) stem from the implications their annotations have on the classes of the entity denoted by the mention: once these classes, i.e., $C$, and the constraints they obey, i.e., $K$, are known for an NLP task $i$, the knowledge of the remaining annotations adds no information. Assumption ($\text{ci}_2$) captures the understanding that the relevant background knowledge for an *individual* NLP task $i$ has been already included in $B$, making the knowledge of $K$ irrelevant for that task, if considered individually. Its implications are that JPARK may be applied only with multiple NLP tasks, being unable by construction to improve a single NLP task when considered alone. Assumption ($\text{ci}_3$) serves to avoid modeling the dependency of $C$ on $m$, and thus $m$ internal structure (a task that we leave to NLP tools). It is a necessary simplification, as in general there is more information in $m$ useful for predicting $C$ than what can be conveyed by annotations $a_i$.

We then consider a single NLP task $i$, and leverage ($\text{ci}_2$), ($\text{ci}_3$), and the definition of conditional probability to express:

$$\begin{aligned}\mathrm{P}(a_i, C|m, B, K) &= \mathrm{P}(a_i|m,B,K) \cdot \mathrm{P}(C|a_i,m,B,K) \\ &= \mathrm{P}(a_i|m, B) \cdot \mathrm{P}(C|a_i, K)\end{aligned} \tag{1}$$

where $\mathrm{P}(C|a_i, K)$ can be learned from a training corpus providing classes (e.g., via links to a knowledge base) and ground truth values of $a_i$ for annotated mentions, as we will describe for NERC and EL in Section 3. $\mathrm{P}(C|a_i, K) = 0$ for all the classes $C$ not consistent with the ontological constraints (e.g., subclass and disjoint axioms) in $K$, and also for many incompatible $\langle C, a_i\rangle$ pairs, meaning that the $C$ to consider are not exponential in the number of ontological classes. We define $\mathcal{C}(a_i, K) = \{C|\, \mathrm{P}(C|a_i, K) > 0\}$ the set of $C$ compatible with $a_i$, and we expect the cardinality of $\mathcal{C}(\cdot, \cdot)$ to be upper bounded by some small constant $c$.

Based on (1), we can derive $\mathrm{P}(C|m, B, K)$ by marginalizing over all the values $a_i \in A_i$ of an arbitrary annotation $i$, or better by averaging (geometric mean) over all the possible marginalizations for $i = 1 \dots n$, to improve the estimate:

$$\mathrm{P}(C|m, B, K) = \left(\prod_i \sum_{a_i \in A_i} \mathrm{P}(a_i, C|m, B, K)\right)^{\frac{1}{n}} \tag{2}$$

Thanks to the independence assumption ($\text{ci}_1$), we can formulate the discriminative model of JPARK as:

$$\begin{aligned}\mathrm{P}(\boldsymbol{a}, C|m, B, K) &= \mathrm{P}(C|m,B,K) \cdot \mathrm{P}(\boldsymbol{a}|m,B,K,C) \\ &= \mathrm{P}(C|m, B, K) \cdot \textstyle\prod_i \mathrm{P}(a_i|m, B, K, C) \\ &= \dfrac{\prod_i \mathrm{P}(a_i, C|m, B, K)}{\mathrm{P}(C|m, B, K)^{n-1}}\end{aligned} \tag{3}$$

where $C$ can be marginalized out by summing over all its possible values compatible with annotations $\boldsymbol{a}$, to obtain:

$$\mathrm{P}(\boldsymbol{a}|m, B, K) = \textstyle\sum_{C \in \bigcap_i \mathcal{C}(a_i,K)} \mathrm{P}(\boldsymbol{a}, C|m, B, K) \tag{4}$$

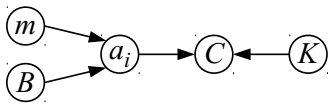


Figure 1: Bayesian network corresponding to Eq. 1. The relations between $m$, $B$, and $K$ are not modeled as irrelevant for our model.

Eq. (2), (3), (4) provide the posterior probabilities of $\boldsymbol{a}$ and/or $C$ (jointly or separately) when the ontological knowledge $K$ is taken into account, and are expressed exclusively in terms of confidences $\mathrm{P}(a_i|m, B)$ provided by the NLP tools, and probabilities $\mathrm{P}(C|a_i, K)$ learned from training data. These equations can be used as such to estimate the *posterior confidences* for a given combination of annotations and/or classes, with time complexities $\mathcal{O}(\sum_i |A_i|)$ for (2) and (3), and $\mathcal{O}(c \cdot \sum_i |A_i|)$ for (4). Alternatively, they can be used to estimate the *optimal annotations and/or classes* — e.g., via $\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} \mathrm{P}(\boldsymbol{a}|m, B, K)$ — with time complexities $\mathcal{O}(c \cdot \sum_i |A_i|)$ for the optimization task based on (2), and $\mathcal{O}(c \cdot n \cdot \prod_i |A_i|)$ for the tasks based on (3) and (4).

Note, finally, that while we did not derive the model graphically, Eq. (1) for the case of an individual annotation $i$ is compatible with the Bayesian network shown in Figure 1.

## 3 NERC + EL Scenario

We instantiate the general model of Section 2 to the specific scenario where mentions are jointly annotated with EL annotations ($a_{\text{EL}}$) and NERC annotations ($a_{\text{NERC}}$), by selecting the necessary ontological knowledge and learning the model probabilities $\mathrm{P}(C|a_{\text{EL}}, K)$ and $\mathrm{P}(C|a_{\text{NERC}}, K)$.

**Ontological Knowledge** As $K$ we use YAGO enriched with the number of ingoing Wikipedia links to the page of an entity, used as a proxy for its number of mentions in Wikipedia (leveraged for estimating priors in (7)). We materialize, applying $\text{RDF}_{\text{pro}}$ [Corcoglioniti *et al.*, 2015], all the inferable classes for an entity based on YAGO TBox (e.g., subclass axioms), obtaining class information for 6,016,695 entities taken from a taxonomy of 568,255 classes.

**Estimating EL Parameters** Entity Linking (EL) is the task of aligning an entity mention in a text to its corresponding entity in a knowledge base $K_{\text{EL}}$. We consider the common case where $K_{\text{EL}}$ is DBpedia, and thus $a_{\text{EL}}$ refers to a DBpedia entity. Since DBpedia and YAGO entities are aligned (via associated Wikipedia pages), the classes for $a_{\text{EL}}$ can be deterministically obtained by mapping $a_{\text{EL}}$ to the corresponding YAGO entity having classes $C_K(a_{\text{EL}})$ in $K$.[5] We thus set:

$$\mathrm{P}(C|a_{\text{EL}}, K) = \mathbf{1}_{\{C_K(a_{\text{EL}})\}}(C) \tag{5}$$

This estimate assumes that $K$ contains complete information about entity classes (closed-world assumption), which usually holds for the most general classes in the class taxonomy.

**Estimating NERC Parameters** Named Entity Recognition and Classification (NERC) is the task of labeling mentions in a text that refer to named things such as persons, organizations, etc., and choosing their type $a_{\text{NERC}}$ from a predefined set of types (e.g., $A_{\text{NERC}} = \{\text{PER}, \text{ORG}, \text{LOC}, \text{MISC}\}$). Since no NERC information is contained in $K$, we assume the availability of a *gold standard corpus* $G$ containing entity mentions annotated with (i) NERC types $a_{\text{NERC}}$ and (ii) ontological classes $C$ aligned with $K$, or alternatively EL annotations deterministically alignable to classes $C$ in $K$.[6] Denoting with $n_G(C, a_{\text{NERC}})$ the number of mentions in $G$ annotated with $C$ and $a_{\text{NERC}}$, and with $\mathfrak{C}$ the set of all class sets in $K$, an estimate of $\mathrm{P}(C|a_{\text{NERC}}, K)$ from $G$ may be:

$$\mathrm{P}(C|a_{\text{NERC}}, G) = \frac{n_G(C, a_{\text{NERC}})}{\sum_{C' \in \mathfrak{C}} n_G(C', a_{\text{NERC}})} \tag{6}$$

Since gold standards are typically small, many combinations $C$ may be observed few times or none at all, and we cannot learn their conditional probabilities. We thus introduce a prior probability of $C$ given only the ontological knowledge $K$:

$$\mathrm{P}(C|K) = \frac{n_K(C)}{\sum_{C' \in \mathfrak{C}} n_K(C')} \tag{7}$$

where $n_K(C)$ is an estimate of the number of mentions in Wikipedia of entities with classes $C$, computed by summing the number of ingoing Wikipedia links of all entities in $K$ with classes $C$. Consequently, we estimate $\mathrm{P}(C|a_{\text{NERC}}, K)$ as follows, where $\alpha$ is a model hyperparameter:

$$\mathrm{P}(C|a_{\text{NERC}}, K) = \alpha \cdot \mathrm{P}(C|K) + (1-\alpha) \cdot \mathrm{P}(C|a_{\text{NERC}}, G) \tag{8}$$

Similarly to EL, also this estimate builds on a closed-world assumption for $K$, mitigated however by the use of priors.

**Restricting the Ontological Classes** As seen above, the limited amount of gold standard data implies that there is little benefit in considering rarely observed ontological classes, even when a prior is introduced. We thus restrict our attention to popular classes of a set $\mathcal{C} = \{c_j\}$ and filter out the remaining classes from $K$. Given a gold standard $G$ (e.g., the NERC one) and denoted with $n_G(c_j)$ the number of mentions in $G$ annotated with ontological class $c_j$ (and possibly other classes), we define $\mathcal{C}$ as the set of all $c_j$ such that:

- $n_G(c_j) \geq \bar{n}$, with $\bar{n}$ a model hyperparameter;
- there is some mention in $G$ not annotated with $c_j$;
- there is no $c_l \in \mathcal{C}$ with $c_l \neq c_j$ and $c_l$ associated to the same mentions as $c_j$.

## 4 Evaluation

The evaluation reported in this paper aims at understanding the potential of JPARK in improving a posteriori, in a scenario where multiple NLP analyses are run, the performances of the NLP tools used. In particular, we consider the NERC and EL scenario of Section 3. Below we describe the tools, datasets, evaluation method, and findings.

[5] In general, a mapping between $a_{\text{EL}}$ and $C$ can be derived via the ontology alignments between entities and classes of $K_{\text{EL}}$ and $K$, and/or a text corpus with EL annotations against both $K_{\text{EL}}$ and $K$.

[6] The approach here described is not specific to EL but can be applied for any annotation $i$ for which a gold standard with $a_i$ annotations and $C$ classes (directly or indirectly supplied) is available.

### 4.1 Tools

To perform NERC and EL analyses, we exploited two state-of-the-art NLP tools for these tasks.

**Stanford NER** [Finkel *et al.*, 2005] This reference tool for NERC provides different models for classifying named entities using different type sets. In our scenario, we exploited Stanford NER with the traditional 4-types CoNLL 2003 model, consisting of NERC types: Location (LOC), Person (PER), Organization (ORG), and Miscellaneous (MISC). Besides returning the best labeling of a sentence, Stanford NER can be instructed to provide many alternative weighted sentence labelings, from which it is possible to derive $a_{\text{NERC}}$ candidates with their confidences $\mathrm{P}(a_{\text{NERC}}|m, B)$ used for JPARK posterior revision of annotations.

**DBpedia Spotlight** [Daiber *et al.*, 2013] This reference tool for EL uses DBpedia as the target knowledge base. Typically (*annotate* service), DBpedia Spotlight returns only one disambiguated DBpedia entity for a spotted mention, but it can also be instructed (*candidates* service) to return 10 weighted candidates for a given mention. We use the latter service to produce the $a_{\text{EL}}$ candidates and corresponding $\mathrm{P}(a_{\text{EL}}|m, B)$ confidences needed for applying JPARK.

### 4.2 Datasets

We use three distinct datasets in our evaluation, in order to verify the capability of our approach to generalize over different annotated data. All the three datasets consist of textual documents together with gold-standard annotations for named entity mentions, both for NERC and EL.

**AIDA CoNLL-YAGO** [Hoffart *et al.*, 2011] This dataset consists of 1,393 English news wire articles from Reuters, with 34,999 mentions hand-annotated with named entity types (PER, ORG, LOC, MISC) for the CONLL2003 shared task on named entity recognition, and later hand-annotated with the YAGO2 entities and corresponding Wikipedia page URLs. It is split in three parts: eng.train (946 docs), eng.testa (216 docs), eng.testb (231 docs).

**MEANTIME** [Minard *et al.*, 2016] The NewsReader MEANTIME corpus consists of 480 news articles from Wikinews, in four languages. In our evaluation, we used only the English section and its 120 articles. The dataset, used as part of the SemEval 2015 task on TimeLine extraction, includes manual annotations for named entity types (only PER, ORG, LOC) and DBpedia entity links.

**TAC-KBP** [Ji *et al.*, 2011] Developed for the TAC KBP 2011 Knowledge Base Population Track, this dataset consists of 2,231 English documents, including newswire articles and posts to blogs, newsgroups, and discussion fora. For each document, it is known that all the mentions of one or a few *query* entities can be linked to a certain Wikipedia page and to a specific NERC type (only PER, ORG, LOC), giving rise to a (partially) annotated gold standard for NERC and EL.

### 4.3 Research Question and Evaluation Measures

In our evaluation, we address the following research question:

**RQ** Does the JPARK a posteriori joint revision of the annotations provided by Stanford NER and DBpedia Spotlight, performed leveraging YAGO ontological knowledge, improve their NERC and EL performances?

By construction, we remark that JPARK relies on the mentions detected by the NLP tools used, so the model may revise the NERC types returned by Stanford NER and/or the EL entities proposed by DBpedia Spotlight, but does not alter other aspects (e.g., mention boundaries). Therefore, to meaningfully evaluate the contribution of JPARK, we consider the following three measures, typically adopted in NERC and EL evaluation campaigns:

- type: a mention is counted as correct if it has the same span and NERC type as a gold annotation. It is the measure used in the CoNLL2003 NER evaluation, and corresponds to strong_typed_mention_match in the TAC-KBP official scorer;[7]
- link: a mention is counted as correct if it has the same span and EL entity as a gold annotation. It corresponds to strong_link_match in the TAC-KBP official scorer;
- type+link: an entity mention is counted as correct if it has the same span, NERC type, and EL entity as a gold annotation. It corresponds to strong_typed_link_match in the TAC-KBP official scorer.

For evaluating the performance on these measures, we use the standard metrics, namely precision $(P)$, recall $(R)$, and $F_1$, computed using the TAC-KBP official scorer on the predicted and gold standard annotations. More in details: true positives $(TP)$ are predicted annotations that are in the gold standard; false positives $(FP)$ are predicted annotations which are not in the gold standard; false negatives $(FN)$ are gold standard annotations which are not among the predicted ones; $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$ and $F_1 = \frac{2 \cdot P \cdot R}{P+R}$.

### 4.4 Evaluation Procedure

We use AIDA eng.train as the gold standard $G$ for estimating the probabilities $\mathrm{P}(C|a_{\text{NERC}}, K)$ of JPARK (probabilities $\mathrm{P}(C|a_{\text{EL}}, K)$ are estimated directly from YAGO) and we use AIDA eng.testa to optimize the model hyperparameters of Section 3, i.e., $\bar{n}$ (best value = 1000, corresponding to 54 YAGO classes and 2041 class sets) and $\alpha$ (best value = 0.02). The evaluation is separately conducted on three datasets: AIDA eng.testb, MEANTIME and TAC-KBP. Note that we do not perform any dataset-specific tuning, and thus the model does not exploit the fact that in MEANTIME and TAC-KBP there are no MISC annotations.

All datasets are automatically preprocessed in order to use entity URIs from the same version of DBpedia (namely, 2016-04) used by DBpedia Spotlight. In particular, the Wikipedia URLs in AIDA and TAC-KBP are aligned to the 2016-04 DBpedia URIs by leveraging the "Redirects," "Revision URIs," and "Wikipedia Links" DBpedia datasets.

The experiment is conducted computing and comparing the metrics for the considered measures in two settings, without (*standard*) and with (*with JPARK*) the contribution of

[7] https://github.com/wikilinks/neleval

| dataset | setting | type | | | link | | | type+link | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| AIDA (5616) | *standard* | 94.30% | 87.50% | 90.80% | 66.20% | 65.20% | 65.60% | 63.40% | 62.50% | 63.00% |
| | *with JPARK* | 95.00% | 88.10% | 91.40% | 67.10% | 65.40% | 66.20% | 65.50% | 63.70% | 64.60% |
| | Δ | **0.70%** | **0.60%** | **0.60%** | **0.90%** | **0.20%** | **0.60%** | **2.10%** | **1.20%** | **1.60%** |
| MEANTIME (792) | *standard* | 88.20% | 69.50% | 77.70% | 70.30% | 55.60% | 62.10% | 63.50% | 50.20% | 56.10% |
| | *with JPARK* | 91.40% | 72.00% | 80.50% | 70.50% | 55.70% | 62.20% | 67.00% | 53.00% | 59.20% |
| | Δ | **3.20%** | **2.50%** | **2.80%** | 0.20% | 0.10% | 0.10% | **3.50%** | **2.80%** | **3.10%** |
| TAC-KBP (4969) | *standard* | 91.10% | 65.20% | 76.00% | 40.10% | 42.30% | 41.20% | 36.70% | 38.60% | 37.60% |
| | *with JPARK* | 92.60% | 66.30% | 77.20% | 41.20% | 42.60% | 41.90% | 38.90% | 40.20% | 39.50% |
| | Δ | **1.50%** | **1.10%** | **1.20%** | **1.10%** | 0.30% | **0.70%** | **2.20%** | **1.60%** | **1.90%** |

Table 1: Precision, recall, and $F_1$ scores for type, link, and type+link measures for both settings on the three evaluation datasets (# of gold standard mentions in parentheses). Score differences (*with JPARK* minus *standard*) are reported, with statistical significant results in bold.

JPARK.[8] More precisely, in the *standard* setting we annotate the documents of the three corpora directly using the highest confidence score NERC type and EL entity proposed by Stanford NER and DBpedia spotlight, respectively. Instead, in the *with JPARK* setting, JPARK picks, among all the candidate annotations returned by Stanford NER and DBpedia Spotlight on the same mention, the NERC type and EL entity (if any) that maximize the joint annotation probability in (4). Note that DBpedia Spotlight returns only 10 candidates, and they may not contain the correct one, or even all of them may be incompatible with all the possible NERC assignments, which is reflected in the posterior probability being zero for all the considered combinations. In these few cases, we assume that the correct EL entity is not among the suggested ones, and thus we drop the EL annotation and keep only the NERC one.

We remark that, as our goal is to understand whether the JPARK posterior revision of the annotations can improve the NERC and EL performances, we focus our study on comparing the scores between the two aforementioned settings, rather than analyzing the absolute scores obtained, which inherently depend also on the performances of the tools providing the candidates. For this reason, no comparative performance evaluation with other state-of-the-art work is reported. Furthermore, as one of the dataset (TAC-KBP) is partially annotated, we consider only the mentions detected by the tools (i.e., annotated with NERC and/or EL) — which are the same in *standard* and *with JPARK* settings — that are in the gold standard, in order to better compare performances across the different datasets, and to avoid obtaining $P$ and $F_1$ scores overly biased by $FP$ in both settings.

### 4.5 Results and Discussion

Table 1 reports precision, recall, and $F_1$ for the evaluation measures on all the datasets, for both settings considered.

For all the metrics computed over the three datasets, the scores are consistently higher in the *with JPARK* setting than in the *standard* one, with improvements ranging from 0.10% to 3.50%. Most of the improvements (23 out of 27) are statistically significant ($p < 0.05$) according to the Approximate Randomization test. Similar outcomes are observed when: (i) considering all mentions returned by the tools (rather than just those in the gold standard), with improvements ranging from 0.10% to 2.80%; (ii) macro-averaging by document, with improvements up to 3.10%; and (iii) macro-averaging by NERC type, with improvements from 0.70% to 2.10%.[9]

Improvements for type+link (from 1.20% to 3.50%), besides being all statistically significant, are always higher than the ones for the other two measures (type and link), thus confirming that the model is particularly effective in jointly selecting the correct $\langle a_{\text{NERC}}, a_{\text{EL}} \rangle$ combination among the available candidate annotations on a given mention.

Analyzing more in detail the results, we can see that the improvement contributed by JPARK on precision is always greater or equal than the one on recall. It indicates that the model, beside fixing annotations (i.e., replacing a wrong annotation with a correct one), also drops, in a few cases, some wrong EL annotations (thus decreasing $FP$).

Separately looking at the results on each dataset, we can see that on AIDA the improvements for type and link are comparable. Instead, on MEANTIME and TAC-KBP, the instantiated JPARK model is substantially more effective for type than for link. While many factors may contribute to this result and further investigations are needed, a possible explanation is that JPARK as used in the experiment is trained on the train part of AIDA, which is likely more similar to AIDA test part than the MEANTIME and TAC-KBP datasets. At the same time, it is worth remarking that the model used for the evaluation, while trained only on AIDA train part, performs reasonably well also on the other datasets, as confirmed by the higher scores for the *with JPARK* setting over *standard* one, with statistical significant improvements in most cases.

Summing up, the results on multiple datasets show that exploiting JPARK for the posterior revision of the annotations performed by Stanford NER and DBpedia Spotlight allows a consistent improvement of their NERC and EL performances, and thus we can positively answer our research question. Furthermore, the positive results obtained over three different datasets with the same instantiation of the model, suggest that

[8] We implemented JPARK as a Java module of PIKES [Corcoglioniti *et al.*, 2016], an open-source knowledge extraction framework exploiting several NLP tools, including Stanford NER and DBpedia Spotlight.

[9] Full results and evaluation material available at http://pikes.fbk.eu/jpark.html

the model may generalize well over different document collections. Together with the generality of the training procedure that needs only NERC and EL gold standards, this suggests that the model may constitute a concrete, ready-to-use solution to (jointly) improve NERC and EL performances.

## 5 Related Work

The contribution presented in this paper may be related to two streams of works: (i) approaches that aim to improve NLP annotations by combining multiple analyses, and (ii) approaches for Knowledge Graph construction from text.

**NLP Annotation Improvement** Previous works have pursued the improvement of performances for some NLP tasks by combining related analyses, mainly NERC and EL.

Some works (e.g., [Stern *et al.*, 2012; Plu *et al.*, 2015]) have proposed pipeline approaches, where named entities are firstly recognized and used to influence the entity disambiguation step. In these approaches, one analysis (NERC) influence the other (EL), but not the other way around.

Other approaches have investigated the development of joint models, exploiting features for multiple tasks and their interactions. For joint NERC and EL models, applied frameworks include re-ranking mechanisms [Sil and Yates, 2013], conditional random field extensions [Luo *et al.*, 2015], semi-Markov structured linear classifiers [Leaman and Lu, 2016], and probabilistic graphical models [Nguyen *et al.*, 2016]. In addition to NERC and EL, coreference analysis has also been considered in the joint model implemented as a structured conditional random field in [Durrett and Klein, 2014].

The contribution presented in this paper differs from all these approaches for several aspects. First, it works a posteriori on the output of existing NLP tools, and thus it is potentially applicable to different tools without altering their own training models or implementations. Second, differently from other approaches (e.g., [Stern *et al.*, 2012; Plu *et al.*, 2015]), our solution does not constrain a directionality on the influence between the considered tasks. Third, while some other approaches (e.g., [Sil and Yates, 2013]) have exploited some background knowledge resource for training their model, though mainly to take into account aspects related to EL, in our approach all annotations are indistinctly mapped to a common ontological knowledge, which plays a central role in capturing the relation and coherence between different annotations on the same mention. We are not aware of other initiatives leveraging some ontological knowledge for this particular purpose.

**Knowledge Graph Construction** The problem of assessing the coherence of different NLP annotations in a Knowledge Extraction context may be related to ongoing initiatives aiming at the construction of Knowledge Graphs from text. In this scenario, the correctness of large sets of potentially noisy ⟨subject, predicate, object⟩ triples, obtained running multiple tools (called *extractors*) on various source types (e.g., documents, HTML pages, spreadsheets), has to be determined.

Typically, approaches tackling this task (e.g., Google's Knowledge Vault [Dong *et al.*, 2014] and DeepDive [De Sa *et al.*, 2016]), derive a truthfulness probability for each extracted triple, obtained considering factors such as its number of occurrences and the quality of extractor and source.

Ontological knowledge is additionally exploited in some approaches to constrain the selection of the extracted candidate triples. A notable example is NELL (Never-Ending Language Learning) [Mitchell *et al.*, 2015], where the strict constraints defined in NELL's ontology (e.g., a person cannot be a city) are used to filter the extracted triples. Other approaches integrate ontological knowledge directly in a probabilistic model, so to jointly consider the ontological constraints and confidence values of the candidates when distilling a Knowledge Graph from extracted triples. Among them, Markov Logic Networks (MLN) are exploited in [Jiang *et al.*, 2012], while Probabilistic Soft Logic (PSL) is proposed in [Pujara *et al.*, 2013]. A different strategy is adopted in SOFIE [Suchanek *et al.*, 2009], where ontological constraints and extracted triples are fed to a weighted MAX-SAT algorithm, whose goal is to select high confidence triples that maximize the number of satisfied constraints.

The problem and the proposed solution here considered differ from all these works for several aspects. First, the above approaches work at the knowledge level on the set of triples typically returned by relation (including entity typing) extractors, and their goal is to select which of the extracted triples to keep in order to be compliant with or to maximize satisfaction of the given set of ontological constraints. Instead, in our work we are interested in working at the level of "generic" NLP annotations, i.e., not necessarily a relation extraction task, and we are interested in using the ontological background knowledge in order to improve the coherence of the annotations on a given mention, and consequently the performances of the NLP tasks. Second, in these approaches the extraction modules are strictly aligned with the relations and classes defined in the ontology used for constraining the triple selection, while in our setting the mapping of the annotations to the ontological knowledge is actually part of the problem formulation, and probabilistically represented. Summing up, our work is not directly comparable with these approaches although the applicability of some of their techniques (e.g., MLN, PSL) to our task may be worth of investigation.

## 6 Conclusions

We presented a novel probabilistic model for improving the annotation of entity mentions by NLP tools. The model explicitly captures the relations between multiple NLP annotations for an entity mention, the ontological entity classes implied by those annotations, and the background ontological knowledge those classes may be consistent with. Given the confidence scores of the candidate annotations identified by multiple NLP tools on the same textual entity mention, the model can be operationally applied to revise the best annotation choice performed by the tools in light of the coherence of the candidate annotations with the ontological knowledge.

We showed how to instantiate the model in a concrete scenario involving two well-know NLP tasks: NERC and EL. The evaluation, conducted using state-of-the-art tools (Stanford NER, DBpedia Spotlight) with three reference datasets, empirically confirmed the capability of the model to improve

the quality of the annotations of the given tools, and thus their performances on the task they are designed for.

Future work will address different directions: (i) application of JPARK to other NERC and EL tools; (ii) evaluation on additional datasets, such as the EL datasets in Gerbil;[10] and, (iii) further validation of the approach's generality for different NLP analyses, investigating a scenario involving additional tasks. A good first candidate for the latter, to consider together with NERC and EL, is Semantic Role Labeling (SRL), as the role an entity can play in a semantic frame is related to the ontological classes for the entity.

## Acknowledgments

The authors thank Yiqing Liang for having contributed to a preliminary implementation of the work.

## References

[Corcoglioniti *et al.*, 2015] F. Corcoglioniti, M. Rospocher, M. Mostarda, and M. Amadori. Processing billions of RDF triples on a single machine using streaming and sorting. In *Proc. of SAC*, 368–375, 2015.

[Corcoglioniti *et al.*, 2016] F. Corcoglioniti, M. Rospocher, and A. Palmero Aprosio. Frame-based ontology population with PIKES. *IEEE Trans. Knowl. Data Eng.*, 28(12):3261–3275, 2016.

[Daiber *et al.*, 2013] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proc. of I-SEMANTICS*, 2013.

[De Sa *et al.*, 2016] C. De Sa, A. Ratner, C. Ré, J. Shin, F. Wang, S. Wu, and C. Zhang. DeepDive: Declarative knowledge base construction. *SIGMOD Rec.*, 45(1):60–67, 2016.

[Dong *et al.*, 2014] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion. In *Proc. of KDD*, 601–610, 2014.

[Durrett and Klein, 2014] G. Durrett and D. Klein. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490, 2014.

[Finkel *et al.*, 2005] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL*, 363–370, 2005.

[Hoffart *et al.*, 2011] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proc. of EMNLP*, 782–792, 2011.

[Hoffart *et al.*, 2013] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[10] http://aksw.org/Projects/GERBIL.html

[Ji *et al.*, 2011] H. Ji, R. Grishman, and H. Dang. Overview of the TAC2011 Knowledge Base Population track. In *Proc. of TAC*, 2011.

[Jiang *et al.*, 2012] S. Jiang, D. Lowd, and D. Dou. Learning to refine an automatically extracted knowledge base using Markov Logic. In *Proc. of ICDM*, 912–917, 2012.

[Leaman and Lu, 2016] R. Leaman and Z. Lu. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, 32(18):2839–2846, 2016.

[Luo *et al.*, 2015] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie. Joint named entity recognition and disambiguation. In *Proc. of EMNLP*, 879–888, 2015.

[Minard *et al.*, 2016] A. Minard, M. Speranza, R. Urizar, B. Altuna, M. van Erp, A. Schoen, and C. van Son. MEANTIME, the NewsReader multilingual event and time corpus. In *Proc. of LREC 2016*, 2016.

[Mitchell *et al.*, 2015] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-Ending Learning. In *Proc. of AAAI*, 2302–2310, 2015.

[Nguyen *et al.*, 2016] D. B. Nguyen, M. Theobald, and G. Weikum. J-NERD: joint named entity recognition and disambiguation with rich linguistic features. *TACL*, 4:215–229, 2016.

[Plu *et al.*, 2015] J. Plu, G. Rizzo, and R. Troncy. A hybrid approach for entity recognition and linking. In *Proc. of Semantic Web Evaluation Challenges ESWC 2015 - Revised Selected Papers*, volume 548, 28–39, 2015.

[Pujara *et al.*, 2013] J. Pujara, H. Miao, L. Getoor, and W. W. Cohen. Knowledge graph identification. In *Proc. of ISWC*, 542–557, 2013.

[Sil and Yates, 2013] A. Sil and A. Yates. Re-ranking for joint named-entity recognition and linking. In *Proc. of CIKM*, 2369–2374, 2013.

[Stern *et al.*, 2012] R. Stern, B. Sagot, and F. Béchet. A joint named entity recognition and entity linking system. In *Proc. of HYBRID*, 52–60, 2012.

[Suchanek *et al.*, 2009] F. M. Suchanek, M. Sozio, and G. Weikum. SOFIE: A self-organizing framework for information extraction. In *Proc. of WWW*, 631–640, 2009.

[Vossen *et al.*, 2016] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A. Minard, A. Palmero Aprosio, G. Rigau, M. Rospocher, and R. Segers. NewsReader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Knowl.-Based Syst.*, 110:60–85, 2016.

[Zadrozny and Elkan, 2002] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proc. of KDD*, 694–699, 2002.