Scalable Understanding of Multilingual MediA (SUMMA)

http://www.summa-project.eu

H2020 Research and Innovation Action
Number: 688139
D4.1 – Initial Progress Report on Automatic Knowledge Base Creation

Nature	Report	**Work Package**	WP4
Due Date	31/07/2017	**Submission Date**	31/07/2017
Main authors	Abiola Obamuyide (USFD), Andreas Vlachos (USFD), Jeff Mitchell (UCL), David Nogueira (PRIB)		
Co-authors	Sebastian Riedel (UCL), Filipe Aleixo (PRIB), Samuel Broscheit (PRIB), André Martins (PRIB), Mariana Almeida (PRIB), Sebastião Miranda (PRIB), Afonso Mendes (PRIB), Andrei Popescu-Belis (IDIAP)		
Reviewers	Shashi Narayan (UEDIN)		
Keywords	Knowledge Base Creation, Information extraction, Named Entity Linking		
Version Control			
v0.1	**Status**	Draft	21/07/2017
v0.2	**Status**	Reviewed	26/07/2017
v1.0	**Status**	Final	31/07/2017

Contents

List of Figures

1 Introduction

1.1 Objectives

(This text taken from the proposal)

This work package takes as input text in multiple languages from WP3 and outputs an automatically constructed knowledge base to be used in the context of media monitoring. Key to the success of this package will be the development of methods that allow the extension of knowledge bases to new relations across languages and the modeling of the temporal nature of relations. The main goals of WP4 are:

- the development of accurate multilingual entity recognition, linking and coreference resolution
- the extension of knowledge base construction to incorporate new relations
- temporal modeling of the relations in the constructed knowledge base

1.2 Background

Considerable effort has been applied to the task of putting common-sense and expert knowledge into digital form (Lenat, 1995; Suchanek et al., 2007; Bollacker et al., 2008b) and these knowledge bases (KB) play an increasingly important role in supporting professional and leisure activities. For example, from 2012, Google began enhancing search results with relevant information from its knowledge graph, while BBC has been maintaining knowledge bases to support journalists for decades.

At an abstract level, a knowledge base is a just a dataset that captures the known facts about a set of entities. However, in practice the particular structures taken by such KBs can differ significantly.

Much academic research has focused on the automatic construction of KBs containing triples consisting of a relationship between a pair of entities, e.g. ⟨`entity1 relationship_to entity2`⟩, where the relationships are drawn from a fairly limited set (e.g. `born_in`, `parent_of`, `owns`). We can think of such a KB as a network where the nodes are entities (i.e. people, places, organisations, etc.) and edges are the relationships between these entities.

In contrast, commercial knowledge bases, such as the BBC Monitoring knowledge base, are often much less structured, for example consisting of short textual summaries for a set of entities. Although, this appears, at least initially, to be a fundamentally distinct structure to the triples described above, some research has begun to look at KBs that contain both types of data. For example, recent Text Analytics Conference Knowledge Base Population (TAC-KBP, Surdeanu and Ji (2014)) evaluations have included triples that contain a free text string, such as *assault with a deadly weapon*, in the place of one entity in the relation *charged_with*.

Historically, both types of data structures have been maintained manually, i.e. by qualified researchers. However, advances in NLP have opened the possibility of automatic construction.

Two slightly different tasks within this category can be identified. The first, knowledge base completion (Bordes et al., 2013; Yang et al., 2015), tries to predict missing links in a KB, e.g. if ⟨`John_Smith works_in London`⟩ is a valid triple then ⟨`John_Smith lives_in London`⟩ is

probably also. Looking at this problem from the perspective of relationships being edges connecting entities within a network, then what is required is a method for predicting missing links in this structure. This sort of analysis is usually based on a statistical analysis of the global network structure to identify reliable correlations which can be used to make such link predictions.

The second task, knowledge base population, attempts to augment a knowledge base with triples derived from text sources, e.g. the text *John Smith, resident of Barnett in London, was remanded in custody last Thursday after ...* supports the triple ⟨`John_Smith lives_in London`⟩. This translation from a textual expression to a KB triple can be carried out locally on a case-by-case basis, particularly if we have an explicit list of textual patterns that reliably predict a KB relation between mentioned entities (Agichtein and Gravano, 2000). Even in the case of a more complex model that predicts a KB relation from features of the text linking the two entity mentions (Nguyen et al., 2007) the predictions would still be generated locally from individual surface patterns, rather than from a global analysis.

Universal Schema (Riedel et al., 2013a), on the other hand, removes the distinction in these models between surface patterns and KB relations by treating both as edges in a network of entities. A factorisation of this graph can then be used to predict new KB entries, in a global manner from both the existing KB relations and also from the observed textual patterns. However, the global nature of this model dissolves the explicit association between a given KB relation and its textual support and also requires that the entire graph of all KB relations and surface patterns be factorised every time predictions are to be made.

Riedel et al. (2013a) employ a fairly parsimonious model to score the plausibility of a given entity pair being linked by a particular relation. Entity pairs and relations are embedded in the same vector space and the score function for that link is based on a simple scalar product between vectors. Section 4 considers a more complex convolutional architecture that allows the prediction of scores for links to all entities simultaneously.

Gold standard Knowledge Base Population (KBP) training data is often only available for small domain-specific problems. and a common tactic to overcome this limitation is to leverage the information present in a large KB. Universal Schema (Riedel et al., 2013a) is one such approach and Distant Supervision (Mintz et al., 2009b) is another. In contrast, the cold start problem requires the extraction of triples when the relation only has a limited number of exemplars. This ability to adapt to and incorporate new relations is potentially very useful in a changing news environment. Section 5 describes our Factorization Machine approach to one shot relation extraction, where one shot refers to the learning setup with a single or a snall number of training examples.

Two pre-requisites for automation of KB construction are definitions of what we mean by entities and relations. A key question in this regard is whether we limit ourselves to a pre-defined list of entities or relations - potentially originating from an existing KB, e.g. Wikipedia entities (Ratinov et al., 2011), or Freebase (Bollacker et al., 2008b) relations - or whether we allow open-ended sets - defined by what is seen in the source documents (Angeli et al., 2015). Such an open domain approach promises the possibility of identifying novel entities and relations without manual intervention, but also threatens to impair the utility of the resulting KB by producing noisy and spurious entries. Limitation to a well-defined universe of entities and relations may be necessary to ensure high-quality outputs. Moreover, when extracting entities from text sources that are not annotated, we are faced with the problem of identifying them automatically, by detecting mention surfaces and performing disambiguation between all the possible entity candidates for that mention. Our approach to Named entity Recognition and Linking is outlined in Section 2.

1.3 Summary

This deliverable documents our progress in automated knowledge base creation and the related technologies of entity tagging and linking. These represent the bulk of the work in Work Package 4 in the first half of the project.

Initially we describe our work on named entity tagging and linking across different languages. This was evaluated in the context of TAC-KBP 2016 which has an evaluation setup compatible with the requirements of SUMMA and is a necessary step for the other tasks in the work package. The next section describes our work on timeline extraction, i.e. extracting timestamped events for entities of interest, which was evaluated on newswire text similar to the domain of SUMMA. Following this, we describe how convolutional neural network architectures can be applied to knowledge base completion. We proceed to detail our work on applying factorization machines to the task of learning knowledge base extractors with limited supervision, which is important for the objective of extending knowledge base construction to new relations. We conclude by describing the software infrastructure being developed to support knowledge base population a well as more generic machine comprehension tasks.

The end-user interface to extracted KB triples within SUMMA is still to be investigated. Automatically extracted triples are obviously noisy and contain some level of error. This may not be desired as the value of commercial knowledge bases, particularly for a journalistic use-case, often lies in their reliability and accuracy. However, automation may potentially offer a dataset of scale and breadth that is unachievable using manual processes.

Thus, the products of the KBP component represent novel resources, that may require some thought to use effectively.

One possible direction is to use the noisy KB triples as inputs to the curation of a high quality knowedge base, with proposed facts being verified by hand against the source documents. This is essentially the approach that Google takes in the maintenance of its knowledge graph.

Rather than seeing the extracted triples as subordinate to a hand curated KB, an alternative direction is to see their value in the insight they provide into the cumbersome volume of news text produced globally. For example, the structured form of the extracted triples facilitates the comparison of reporting across sources, enabling the identification of reports that agree with or contradict each other. Equally, extracted facts could be compared to a reference knowledge base.

Both directions would probably contain similar elements. Each will require some means of viewing the extracted triples. For a large set of triples, this is probably as a list, but a smaller number of related triples could also be usefully viewed graphically as a network. The link between triples and their supporting sources also needs to be handled, and this connection is bi-directional: from triples to sources and also from documents to triples. Given proposed facts, a user will want to be able to see the supporting sources, but will also probably want to make the reverse connection to view the extracted facts for a set of documents. Relationships between facts, such as contradiction or entailment, may be another factor of interest to users, requiring some interface.

Thinking about this resource as part of a workflow, users may desire the ability to flag triples as incorrect or unreliable, to avoid viewing them again.

1.4 Deliverable strucure

The present report consists of the following parts:

- The work carried out in the fields of entity tagging and linking, necessary steps for the knowledge population task.
- The development of a timeline extraction approach using joint inference.
- The use of convolutional architectures for link prediction in knowledge bases.
- The application of factorization machines to the task of learning knowledge base extractors with limited supervision.
- The software infrastructure named Jack The Reader (JTR) being developed to support knowledge base population as well as more generic machine comprehension tasks

The first 4 bullet points and related sections discuss improvements in knowledge base population and entity linking and tagging technologies, while the last one is about ongoing work in software integration.

2 Multilingual entity recognition and linking

2.1 Introduction

The task of entity linking, also called named entity linking (NEL) and named entity disambiguation (NED), aims at connecting detected mention occurrences in a document or multiple documents with the known entities they refer to, which are stored in a knowledge base k. The main challenge resides in the fact that entities are not necessarily mentioned with their distinct and most complete knowledge base identifier (entity name) and a query to the knowledge base with an incomplete or ambiguous name can yield multiple entity candidates $c \in k$. Therefore, statistics obtained from large document sets with links to knowledge base identifiers, as well as their contexts needs to be leveraged for an accurate disambiguation.

The task of entity linking is preceded by named entity recognition (NER), which attempts to detect all the named entities listed in a document. Named entity recognition and entity linking are fundamental steps to achieve good performance in downstream applications, such as knowledge base population and search, and question answering systems. Such developed modules will be used extensively in this work package, the clustering task of WP3 and also in WP5. We will take a multilingual approach focusing on the core languages of the project (English, German, Spanish) and for Portuguese.

The entity tagging and linking (ETL) module will be evaluated as a standalone task using well-established datasets, such as the ones distributed for the Text Analysis Conference – Knowledge Base Population (TAC-KBP) [1] shared tasks and the system performance will be measured with the standard metrics that are used for evaluating those datasets, including the correct identification of mentions in text, the correct classification of mentions with entity types (such as specific individual person (PER), organization (ORG), geopolitical entity (GPE), etc) and the correct classification of mentions with entities in a knowledge base.

This section is organised as follows. Section 2.2 states the original description of work, taken from the proposal. Section 2.3 describes the work performed by Priberam in the named entity recognition task, Section 2.4 describes Priberam's devised approaches to tackle the entity linking task, as well as our contribution to the Entity Discovery and Linking (EDL) track in the TAC-KBP competition and in Section 2.5 IDIAP's and Priberam's coreference resolution experiments are presented.

2.2 Original description of work

Task T4.1: Multilingual Entity Recognition, Linking, and Coreference Resolution (PRIB, LETA) (Text taken from the proposal)

This task aims to develop statistical models for entity recognition, entity linking, and coreference resolution. These are fundamental steps toward automatic knowledge base construction and the modules developed will be used extensively in this work package and also in WP5 and in the clustering task of WP3. We will take a multilingual approach for the core languages of the project (English, German, Spanish) and for Portuguese. For the remaining languages, we will use machine translation with preserved word alignments to the original and then revert to an English system. For

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

the named entity recognizer, we will go beyond simple person/organization/location/event entity tags toward more fine-grained information, based on the ontology defined in WP2. For languages for which annotated text with these tags is not available, we will apply robust cross-lingual transfer techniques based on a large pool of parallel data. For the entity linker, we will implement a new approach based on the notion of "entity embeddings." Namely, for every given entity in a knowledge base (for example the name of a football player), we will compute a continuous or latent representation in a low-dimensional vector space; semantically similar entities (e.g. several football players in the same team) will be placed close to each other. The computation of these embeddings will be carried out off-line, by looking at the local graph neighborhood of each entity (e.g. all Wikipedia pages that link to an entity page). When a document is presented in real-time to the entity linker, contextual embeddings for the document are also computed (for example, simply combining word embeddings) and an entity will be picked from the candidate list by taking into account the similarity between the entity embedding and the document embeddings, as well as the importance (e.g. PageRank score of the candidate entity). This will automatically promote consistency among the entities in a document(e.g. disambiguating several football player's to be the ones playing in the same team), while avoiding expensive random walks or other graph operations as prior approaches do. This speed-up is fundamental for practical usage of this system with large volumes of data. In addition, we will output a confidence score about each linked entity. For certain cases, an entity is referred to in the document that is not yet in the knowledge base. In that case, we will run a coreference resolver to cluster all mentions of this entity in the document and automatically add the new entity to the database. The canonical English transliteration for such newly found entities will be generated automatically based on external databases such as DBPedia or Freebase and later can be corrected through manual curation via user-interface. The coreference resolver will also be multilingual and capable of resolving pronouns and nominal mentions. It will also add a confidence score to the found coreferent mentions.

2.3 Named entity recognition

Our contribution to the named entity recognition task is threefold:

- To improve the results of the named entity recognition task, several datasets (TAC [2], OntoNotes [3] and ACE [4]) were preprocessed, we describe them in Section 2.3.1.
- For detecting and labelling mentions, and currently integrated in the SUMMA platform pipeline, we use an enhanced version of the named entity recognition (NER) system available in the TurboParser (Martins and Almeida, 2014) pipeline. These improvements are discussed in Section 2.3.2.
- A novel and more research-oriented approach based on deep conditional random fields (deep-CRFs) is also being developed, which is described in Section 2.3.3.

2.3.1 Pre-processing of NER datasets

The goal of this processing is to improve our performance in the TAC competition. With the intent of obtaining more data for trainning, besides TAC Train (2015) and Dev (2016) datasets, we used

[2] https://tac.nist.gov/
[3] https://catalog.ldc.upenn.edu/docs/LDC2013T19/
[4] https://www.ldc.upenn.edu/collaborations/past-projects/ace

OntoNotes and ACE. Both OntoNotes and ACE were processed to achieve congruence witht the TAC dataset, namely in the name tagging and span boundaries.

Namely, in the OntoNotes dataset, where the used domains were the broadcast news, the newswire and the web data, we shorten span boundaries and mapped spans with "Nationality, or Religious or Political Organization" (NORP) tag to GPE, LOC, ORG or no tag (removed). In the ACE dataset, where the used domains were the broadcast news), newswire, usenet newsgroups/discussion forum and weblog, we selected the shortest spans, removed mentions with tags VEH and WEA and selected named mentions "NAM" only (no nominal mentions, "NOM"). In all datasets, whenever we had nested mentions, we kept the longest one. For the TAC dataset, we used data from the newswire and discussion forum domains. Experiments were both carried out for the English and Spanish domains for the TAC dataset.

For the neural approach, the used pre-trained embeddings were the 300 dimensional ones from GloVe (Pennington et al., 2014), and they were calculated from the Wikipedia 2014 and Gigaword 5 datasets.

2.3.2 TurboParser enhancements

In TurboParser (Martins et al., 2013), the named entity recognition (NER) system implements a linear sequential model whose features are based on the Illinois Entity Tagger (Ratinov and Roth, 2009). The cost function of the Turbo named entity recognition system was adapted to take into account mismatches between predicted and gold mentions, with a cost C_{FN} for mentions that were incorrectly predicted (false negatives) and another cost C_{FP} for gold mentions that were missed (false positives). These two parameters (C_{FN} and C_{FP}) can be tuned in the development set in to adjust the trade off between precision and recall, and achieve the highest F_1 score.

Regarding performance, besides accelerating TurboParser's named entity recognition with code optimisation, that led to a 30% speed performance improvement with a single core usage, execution time was also improved with a continuous effort to maintain a NLP pipeline that allows for multithreaded calls to this module.

Experimental results After tuning the aforementioned parameters to adjust the trade off between precision and recall, the following F1 scores presented in Table 1 and Table 2 were obtained. Previous F1 scores for NER (named entity span detection) and NERC (named entity span detection + type classification) were 83.1% and 76.1%, respectively, as reported in (Paikens et al., 2017).

Table 1: English TurboParser NER Precision, Recall and F1 breakdown, per tag, using the (2) TAC+OntoNotes dataset, on the TAC2016 test dataset (newswires and discussion forums)

	Gold	Pred	OK	Precision	Recall	F1
NER	7009	6518	5796	88.92%	82.69%	85.70%
NERC	7009	6518	5495	84.31%	78.40%	81.24%
PER	2546	2405	2188	90.98%	85.94%	88.39%
GPE	2323	2094	1920	91.69%	82.65%	86.94%
ORG	1679	1760	1193	67.78%	71.05%	69.38%
FAC	112	61	49	80.33%	43.75%	56.65%
LOC	349	198	145	73.23%	41.55%	53.02%

Table 2: Spanish TurboParser NER Precision, Recall and F1 breakdown, per tag, using the (2) TAC+OntoNotes dataset, on the TAC2016 test dataset (newswires and discussion forums)

	Gold	Pred	OK	Precision	Recall	F1
NER	5385	4520	4036	89.29%	74.95%	81.49%
NERC	5385	4520	3862	85.44%	71.72%	77.98%
PER	2103	1980	1804	91.11%	85.78%	88.37%
GPE	2082	1742	1559	89.49%	74.88%	81.54%
ORG	883	692	429	61.99%	48.58%	54.48%
FAC	63	15	4	26.67%	6.35%	10.26%
LOC	254	91	66	72.53%	25.98%	38.26%

2.3.3 Named entity recognition using Deep-CRF

Priberam's efforts in building a neural network based NER system were built upon the model proposed by Ma and Hovy (2016), which uses a combination of Bidirectional Long-Short Term Memory (BiLSTM), Convolutional Neural Network (CNN), and Conditional Random Fields (CRF). This model requires no language specific knowledge, and relies on vectorial representations both from the word and character levels. The main goals of this work were twofold: making a submission to the Text Analysis Conference (TAC) 2017, and exploring the NER capabilities of deep models using semi-supervised training data from different domains with the aim of overcoming the obstacle that the lack of supervised training data imposes.

In order to build on top of the model proposed in Ma and Hovy (2016), we took the TensorFlow Abadi et al. (2015) implementation from Genthial (2017) as a starting point. This implementation has, although, a slight architectural difference from the one in Ma and Hovy (2016), as it uses a BiLSTM in order to obtain the character-level word representation instead of a CNN. We experimented with both BiLSTM and CNN for this purpose. Moreover, we implemented two different ensembling architectures, which will be briefly explained later in this section.

In the architecture proposed in Ma and Hovy (2016), here denoted as the *basic* model, a CNN is first used in order to obtain the representation of each word from its constituent characters, as presented in Figure 1. The character-level word representations are then concatenated with the corresponding pre-trained word embeddings. As depicted in Figure 2, fter combining the character and word-level representations, the result is fed into a BiLSTM, from which the context-level representation of each word in the sentence is obtained. The output tensor of the context-level BiLSTM is, in this report, referred to as the *logits*. On top of the BiLSTM, a sequential CRF is then used to jointly decode the best label sequence. In order to avoid incongruent tagging, hard constraints were imposed on the CRF's transition tensor, such that all of the illogical transitions were prohibited. Moreover, a hard constraint was also imposed on the logits' tensor score for the first word in the sentence, in order to prohibit it from ever being decoded as an *inside* tag.

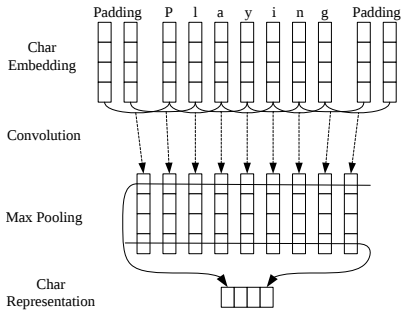

Figure 1: The CNN used to extract word representations from the character level. Dashed arrows indicate a dropout layer applied before the character-level embedding are input into the CNN.Taken from Ma and Hovy (2016)

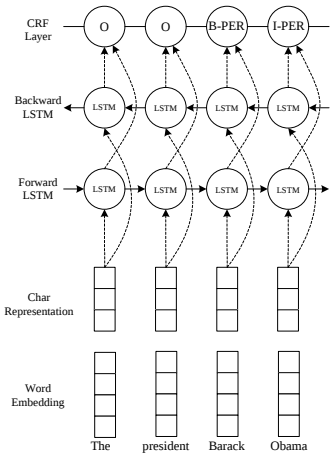

Figure 2: The architecture of the neural system used for NER. The character-level representation of each word is obtained from the CNN depicted in Figure 1. Those word embeddings are then concatenated with pre-trained word embeddings from GloVe Pennington et al. (2014) before being input into the BiLSTM. Dashed arrows indicate dropout layers applied both to the input and to the output of the BiLSTM. Taken from Ma and Hovy (2016)

To go beyond the simple architecture, with the aim of increasing the robustness of the basic model, we implemented an ensembling architecture where we first trained 10 basic models, each of them initialized with a different random seed. We then combined them in order to make predictions on the test set by averaging the logits tensors from the various independently trained models. Finally, we used the average transitions tensor to decode the average logits. As an alternative approach to the traditionally implemented ensemble, where each of the sub-components is separately trained, we experimented with an architecture that might be described as a jointly trained ensemble. In this architecture, we replicated the layer used to obtain the character-level representation and the context BiLSTM layer 5 times. The logits obtained from each of the context BiLSTM layers was then averaged out and given as input to a single CRF layer, such that a single transitions tensor was learned.

Experimental results In this section, we present the results of the experiments that we ran, for the English and Spanish domain. The corpora used for training the model was comprised of data from the TAC, OntoNotes and ACE datasets; different combinations of these datasets and model settings were experimented with. The metrics here presented in Tables 3, 4 and 5 were calculated for the test set of TAC 2016, including both the newswires and the discussion forums, and were obtained after a post-processing step which included the addition of authors and nested mentions.

Table 3: English Deep-CRF NER Precision, Recall and F1 breakdown, per tag, using the (2) TAC+OntoNotes dataset, on the TAC2016 test dataset (newswires and discussion forums)

	Gold	Pred	OK	Precision	Recall	F1
NER	7009	6828	6080	89.05%	86.75%	87.88%
NERC	7009	6828	5846	85.62%	83.41%	84.50%
PER	2546	2574	2372	92.15%	93.17%	92.66%
GPE	2323	2180	2015	92.43%	86.74%	89.50%
ORG	1679	1706	1199	70.28%	71.41%	70.84%
FAC	112	104	72	69.23%	64.29%	66.67%
LOC	349	264	188	71.21%	53.87%	61.34%

Table 4: Spanish Deep-CRF NER Precision, Recall and F1 breakdown, per tag, using the (2) TAC+OntoNotes dataset, on the TAC2016 test dataset (newswires and discussion forums)

	Gold	Pred	OK	Precision	Recall	F1
NER	5385	5214	4374	83.89%	81.23%	82.54%
NERC	5385	5214	4206	80.67%	78.11%	79.37%
PER	2103	2069	1848	89.32%	87.87%	88.59%
GPE	2082	2143	1753	81.80%	84.20%	82.98%
ORG	883	903	544	60.24%	61.61%	60.92%
FAC	63	1	0	0.00%	0.00%	0.00%
LOC	254	98	61	62.24%	24.02%	34.66%

Table 5: Results obtained for the English experiments using different train datasets, on the test set of TAC 2016, including both the newswires and the discussion forums. These experiments were performed using only the basic architecture, with a CNN at the character level and non-trainable PTWE's.

		F1 NER				F1 NERC			
English dataset		Max.	Min.	Mean	Var.	Max.	Min.	Mean	Var.
TAC	Reduced	83.56	81.81	82.58	0.32	79.17	76.49	77.58	0.54
	Final	83.86	83.01	83.37	0.06	79.53	78.53	79.04	0.08
TAC + OntoNotes	Reduced	88.44	87.60	87.96	0.06	84.33	83.27	83.80	0.10
	Final	87.69	87.06	87.35	0.04	84.16	83.35	83.82	0.06
TAC + OntoNotes + ACE	Reduced	88.29	87.11	87.49	0.15	84.06	83.21	83.60	0.08
	Final	87.81	86.65	87.26	0.12	84.71	83.71	83.97	0.08

2.4 Entity linking

Our contribution to the entity linking task is also twofold:

- An entity linking system (Paikens et al., 2017) based on an information retrieval Nearest-Neighbours-assisted ruled-based system (Amaral et al., 2008) was submitted last year to the TAC Knowledge Base Population - Entity Discovery and Linking track, TAC-KBP EDL 2016. Since then we have improved over this version and its performance will be outlined in Section 2.4.1.
- Another entity linking system, described in Section 2.4.2, is an initial implementation of a language independent approach, which was also submitted to TAC-KBP EDL 2016. It is an easy-first search sequence prediction neural network trained with "universal features", namely, features obtained from pre-trained cross-lingual representations (see our work, Ferreira et al. (2016)).

2.4.1 Nearest-neighbours-assisted rule-based entity linking system

In this section, we describe Priberam's nearest-neighbours-assisted ruled-based entity linking system, of which an earlier version was described in Paikens et al. (2017). The mentions detected as reported in Section 2.3 are linked to KB entries according to the rule-based strategy that will be described in the following sections, summarized in Algorithm 1. Section 2.4.1 presents the experimental results.

Algorithm 1 Linking System

1: high-precision sub-string match mention coreference
2: Candidate generation
3: Candidate rank step #0: information retrieval engine (KNN algorithm) + prior statistics
4: Candidate re-rank step #1: accounting for co-occurrences between all mentions candidates
5: Candidate re-rank step #2: accounting for coherence
6: Global NIL clustering and cross-document coherence

Entity linking indexes In entity linking, due to the necessity of linking recognised 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 the desired entities to link to, 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 entity cooccurrence models based on in-link sources (Chisholm and Hachey, 2015). 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.

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 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).

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

In the 4^{th} step, 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-neighbours search engine generated 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, 2006). 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 similarity is computed between the candidates of all mentions, based on co-occurring entities. A certain mention candidate co-occurrence similarity score will be greater whenever that entity co-occurs with more entities from the other candidate mentions and with less ambiguous mentions (mentions with a lower number of candidates).

In the 5^{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 an additional feature, the similarity between co-occurring candidates. Following the reorder by that new similarity score, one last reorder step is applied, accounting for coherence.

Coherence re-rank Contrary to state-of-the-art entity linking methods that favour 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:

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

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 ($s_{j,l}$), and which is given by:

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

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 Eqn. 1, is similar with the model that was independently proposed by Globerson et al. (2016).

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

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

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 Eqn. 3.

Experimental results In this section the system is compared against state-of-the-art technology (Ji et al., 2016) by evaluating on standard metrics and datasets and by comparing with the best competing systems from the TAC-KBP EDL shared task. Such task is very aligned with SUMMA goals both in terms of the objectives and of the languages that are covered. We have already participated in TAC-KBP 2016 where we have scored as an intermediate system at the task of entity discovery and linking (Paikens et al., 2017). From this competition we concluded that, at that stage, the bottleneck of the entity discovery and linking module was the mentions detection system, while the step of linking the mentions to knowledge base entries was on pair with the top

competing systems. Current results show that we would have a system capable of attaining best NERLC (entity mention span detection + type classification + knowledge base ID disambiguation) results for English and Spanish, if we are able to use a mentions detector of equivalent performance as the best one from last year, as seen in Tables 6 and 7. Previous F1 scores for NERLC, KBIDs and CEAFm, applying the ENG-NAM (English Named mentions) filter and using the USTC mentions, were 79.8%, 81.0% and 83.3%, respectively, as in (Paikens et al., 2017).

Table 6: Overall Entity Discovery and Linking Performance (%) for the TAC-KBP 2016 Corpus

F1 scores obtained for the 3 languages, with our current entity linking system and the USTC NELSLIP3, both considering the detected mentions from the USTC NELSLIP3 NER system from TAC 2016 submission, on the LDC2016E63 TAC-KBP 2016 Evaluation Source Corpus, for the following Evaluation measures for entity discovery and linking: NER - strong_mention_match, NERC - strong_typed_mention_match, NERLC - strong_typed_all_match, KBIDs - entity_match, CEAFm - mention_ceaf.

	our system					USTC NELSLIP3 (2016)				
	NER	NERC	NERLC	KBIDs	CEAFm	NER	NERC	NERLC	KBIDs	CEAFm
English	0.818	0.774	**0.670**	**0.776**	**0.710**	0.818	0.773	0.669	0.760	0.699
Spanish	0.766	0.730	**0.684**	**0.761**	**0.731**	0.766	0.730	0.614	0.720	0.656
Chinese	0.749	0.692	0.568	0.675	0.624	0.783	0.743	**0.642**	**0.730**	**0.711**

Table 7: Overall Entity Discovery and Linking Performance (%) for the TAC-KBP 2016 Corpus ({ENG_NAM, SPA_NAM} - named mentions only)

F1 scores, for Named mentions only, obtained with our current entity linking system and the USTC NELSLIP3, on the same dataset, with the same detected named mentions from the USTC NELSLIP3 NER system from TAC 2016 submission and using the same measures as in Table 6.

	our system					USTC NELSLIP3 (2016)				
	NER	NERC	NERLC	KBIDs	CEAFm	NER	NERC	NERLC	KBIDs	CEAFm
English	0.906	0.879	**0.819**	**0.825**	**0.873**	0.906	0.878	0.792	0.811	0.832
Spanish	0.875	0.839	**0.786**	**0.796**	**0.838**	0.875	0.839	0.705	0.757	0.752

2.4.2 Easy-first structured prediction entity linking system

Rationale and pipeline design Approaches to entity mention disambiguation can be classified into either making an independent local decision for each mention (Cucerzan, 2007; He et al., 2013; Yamada et al., 2017), or global methods that additionally exploit the semantic relatedness of entities to promote a coherent configuration (Han et al., 2011; Hoffart et al., 2011; Ganea and Hofmann, 2017). The challenge for global methods is, that the semantic relations between candidate sets yield an exponential search space of possible configurations. Previous work tackled this either by using heuristics or approximate inference. In this work, we model the aspect of finding a coherent configuration as a structured prediction problem, i.e. $C : X \rightarrow Y$, where X are sequences

of mentions $m1, m2, ...$ in a document d, and Y are sequences $e1, e2, ...$ of entities, which yield an exponential search space of combinations during inference. Learning-to-search (Daumé III and Marcu, 2005) tackles this by casting it as the problem of learning a classifier to navigate the search space sequentially. Such a classifier is, at least, based on local features as well as on the history of selected entities H, i.e. $C : X \times H \rightarrow Y$.

Such history of previous actions, given a set of possible entities, represents a set of dense graphs of semantically linked entities. The semantic links are latent, i.e. are not annotated in the training data nor do we necessarily have to label them. In regards to the predicted structure, the closest prior work is from Fernandes et al. (2012) and Björkelund and Kuhn (2014), who learn a classifier that constructs latent antecedent trees for coreference.

Furthermore, we see this as a typical easy-first search problem, based on the observation that less ambiguous mentions can help to solve more difficult ones (Heinzerling et al., 2015). The challenge in developing an easy-first model is, that the "easyness" is not annotated in the training data but has to be learned from the data. Contrary to a left-to-right approach, an easy-first also implies that the learning procedure is fundamentally different, because we learn a classifier that predicts the easiest mention as well as its entity. Xie et al. (2015) summarised and formalised the prior published research for easy-first approaches and derived a generic perceptron style online learning algorithm. We followed such formulation and transferred the concept to a log-linear model and apply it to entity linking. Also, we designed a non-linear architecture that uses distributional representations of entities and mention contexts with the final goal to use multilingual embeddings to make the disambiguation model invariant to a specific language. Another reason to aim at distributional representations for entities is, that they can be learned from data that is annotated by non-experts, i.e. Weblinks to the knowledge base, instead of relying on expertly annotated semantic relations from a knowledge base. Also, in a non-linear architecture we can learn many aspects of natural language processing jointly and do not have to resort to a pipeline solution and hand engineered feature extractors, which has been shown to also be beneficial also for entity linking.

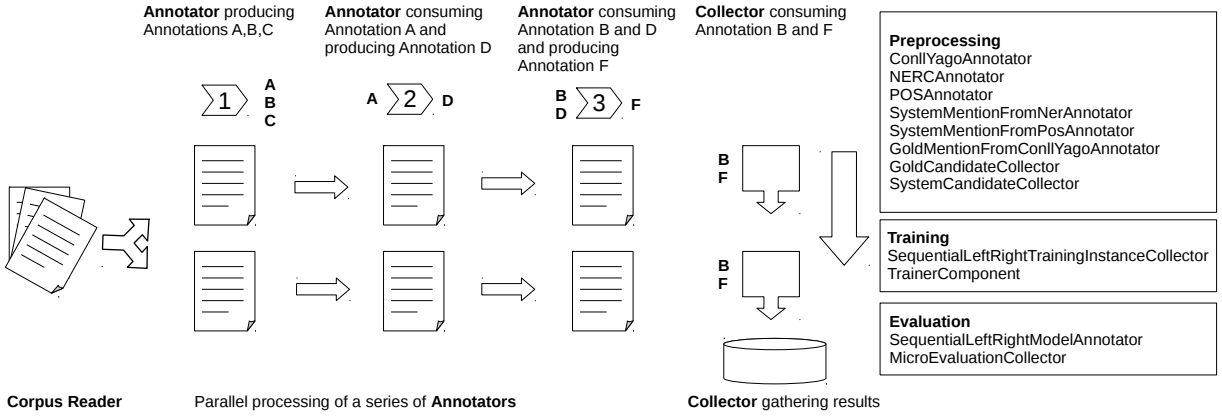

Figure 3: Annotation framework data flow and a pipeline example.

A framework was developed based on the widespread ideas from UIMA (Unstructured Information Management Architecture, Ferrucci and Lally (2004)), and compiler build systems like *make*[5].

[5] https://www.gnu.org/software/make/manual/make.html

Two ideas were borrowed from UIMA: the concept of non-destructive annotation that only creates annotation layers on-top of the source data, and a map-and-reduce-like pipeline architecture. The core idea borrowed from build systems like *make* is the dependency graph of intermediate steps to the final outcome, taken as a principle for the pipeline design.

The map-and-reduce-like pipeline architecture is reflecting the fact that, in text processing, after reading the source data, the documents can usually be processed independently and, therefore, in parallel and are also often processed by several components in series. For example, a classic pipeline consists of several annotation steps, like tagging, parsing, named entity detection and then, as for example in our case, entity linking. This could be finished by several evaluation steps, which have to gather the annotations from all the documents. See Figure 3 for an illustration for this.

Entity linking pipelines with a linear classifier Using this framework, an entity linking model based on two linear classifiers was implemented. For feature extraction, using as features the local feature functions $f(c_{ij}, m)$ that measure the compatibility of the candidate c_{ij} given the mention m_i, and sequential feature functions $f(c_{ij}, h)$ with $h = \{(m_{i-1}, \hat{e}_{i-1}), (m_{i-2}, \hat{e}_{i-2}), ...\}$ that measure the compatibility of the candidate given the history of selected entities within the current document.

The used local features were: mention candidate string similarity $f_{str(m,c)}$; entity prior $f_{prior(e)}$; entity given mention prior $f_{prior(e|m)}$; mention context similarity $f_{ctxtsim}$; and mention prior $f_{prior(m)}$. Sequential feature functions measure the compatibility of one candidate to the history, and since we are interested in the compatibility to the full history, therefore, we always apply some pooling (min,max,average) over the history. The ones used were: mention mention string similarity $f_{str(m,c,h)}$; candidate similarity to pooled history $f_{simpoolhist}$; pooled candidate similarities to history $f_{poolsimhist}$; and entity cooccurence $f_{poolcooc}$.

In contrast to a left-to-right, in the implemented easy-first regime we have no fixed order and we have no gold annotation for the easiest mention-entity pair. In fact, we want to learn to order the mentions by "easyness". Let $S = \mathcal{P}(\{m_1, m_2, ...\})$ be the search space , i.e. the power set of all mentions, with $s \in S$ is a state in the search space, i.e. $s = \{m | m\ is\ pending\}$ are the yet unresolved mentions. $A(s)$ is the set of possible actions for state s, i.e. $A(s) = \{m \times q_{KB,h}(m) | m \in s\}$ are the pairs of mentions and candidate entities, and $h \in H$ is the history of actions made so far.

$$C_{EFLL} : S \times H \rightarrow A$$

This is achieved by learning a scoring function

$$\begin{aligned} score : A \times S \times H &\rightarrow \mathbb{R} \\ score((e,m), s, h) &= \theta^T f((e,m), s, h) \end{aligned}$$

However in our experiments we did not use features based on the pending mentions, therefore our *score* function is defined as

$$score = \theta^T f((e,m), h) = \theta^T f(e, m, h)$$

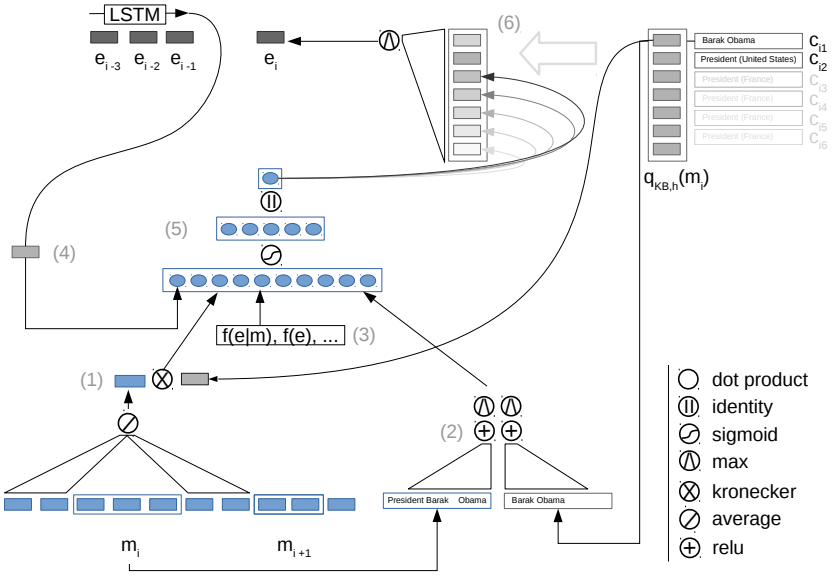

Figure 4: Sketch of neural model. (1) mention context encoder over word embeddings and alignment to entity embedding, (2) convolutions over characters of mention string and candidate string, (3) popularity priors, (4) last hidden layer of LSTM over history of selected entity embeddings, (5) network with one hidden layer, (6) scores for each candidate entity c_{ij} computed by the model.

Non-linear classifier Another approach, in the line of non-linear classifiers, was to use neural networks. A neural model yields the advantage that we can *learn* functions, like, for example, the string compatibility of the mention string and the entity title string. The goal in designing a neural network architecture is, to enable the network to learn those functions and exploit the structure of the input. The only feature functions from aforementioned ones that we are not be able to replace with a learned function are the popularity functions $f_{prior(e|m)}$ and $f_{prior(e)}$.

We want to learn a classifier C_{NNLL} that predicts an entity for a mention m_i from a set of candidates $c_{ij} \in q_{KB,h}(m_i)$ using a function F

$$\hat{e}_i = \underset{c_{ij} \in q_{KB,h}(m_i)}{argmax} \; F(c_{ij}, m_i, h)$$

The network F in Figure 4 computes a score based on c_{ij}, m_i and h, like in the linear model previously described.

Experimental results Early results from this easy-first sequence prediction system with "universal features" show promising results, although not yet comparable in terms of $F1$ with the ones from the Nearest-Neighbours-assisted ruled-based system, described in Section 2.4.1, as can be seen in Table 8. This approach will therefore, continue to be a research direction with the aim of attaining a state-of-the-art performing language-independent entity linking system.

Algorithm	Classifier	Features	$F1_{MI}@1$
Left-To-Right	C_{LL}	FULL	0.8196
Left-To-Right	C_{NNLL}		0.8090
Nearest-Neighbours-assisted ruled-based system			**0.9005**

Table 8: Comparison of the linear model and the non-linear model with full feature set and the left-to-right algorithm on the blind test set (testb) of the AIDA-YAGO CONLL data and comparison to state of the art.

2.5 Coreference resolution

Two approaches to coreference resolution have been studied. The first one is already operational within Priberam's nearest-neighbours-assisted ruled-based entity linking system, and has been described in Section 2.4.1 above. We describe here pilot experiments done at IDIAP with a deep neural network approach, inspired from an existing Long Short-Term Memory architecture embedded with a memory network (hence, LSTMN) (Cheng et al., 2016). While the original model was designed for natural language inference, we studied its application to pronominal anaphora resolution.

To achieve this, we extracted all pronominal mention pairs from the English portion of the CoNLL 2012 coreference shared task data (Pradhan et al., 2012), and structured the data in three columns. The first column contained a binary label for correct or incorrect coreference replacements. The second column has the original sentence from the dataset and with the pronominal anaphor to replace. The third column contains the sentence with a candidate antecedent replacement. Our initial premise was that the intra and inter attention mechanisms would allow the model to detect correct coreference resolutions amongst several sentences with correct or incorrect replacements.

With a parameter count of 3.4M and size of 450 for the hidden layer, the model was able to recognise correct antecedents in 35.3% of test cases. We used the the KenLM language modelling toolkit (from Moses) trained on Europarl to check if the model was learning more than local relationships between words (n-grams). The language model was able to detect 29.9% of correct replacements, with only an 11.7% overlap between the sets of correct answer choices in the two models. This suggests that the LSTMN may be able to identify some of the longer-range constraints that help anaphora resolution.

2.6 Conclusion and future work

We analysed the task of identifying and linking to a knowledge base entity mentions in a collection of documents. We have attained reasonable results with Priberam's entity linking in the TAC-KBP 2016 submission and later developments show promising results. Current results show that we would have a system capable of attaining best NERLC results for English and Spanish, if we manage to improve our mentions detector to a level of performance equivalent or better than the best ones from last year (Ji et al., 2016).

We aim to submit our entity linking to TAC-KBP 2017 Entity Discovery and Linking task later this year. Future work will include improving our mentions detector, a necessary step for a good performance in entity discovery and linking, and improving entity linking processing speed. Regarding integration aspects, TurboParser's named entity recognition and coreference resolution

systems, and Priberam's entity linking are currently deployed on standalone servers and integrated with the rest of the SUMMA platform via a RESTful API, as described in Deliverable D4.2.

2.7 Publications

- 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. Summa at tac knowledge base population task 2016. In *Proceedings of the Text Analysis Conference -TAC*, pages 1–9, Gaithersburg, Maryland USA, 2017

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

3 Timeline extraction using distant supervision and joint inference

3.1 Introduction

Temporal information extraction focuses on extracting relations and events along with the time when they were true or happened. This is an important aspect of knowledge base population for SUMMA, since some of the information compiled by BBC Monitoring Research is timestamped events related to a particular entity, either a person or organization.

In this section we focus on the task of timeline extraction following the recent SemEval TimeLine shared task[6] (Minard et al., 2015). The aim of the task is to extract timelines from multiple documents consisting of events in which a given target entity is the main participant. An example timeline for the entity *Steve Jobs* extracted from 4 documents is given in Fig. 5.

The development data provided by the TimeLine shared task does not contain annotations for the various intermediate processing stages needed, only a set of documents with annotated event mentions (input) and the timelines extracted for a few target entities (output). No training data was provided, thus participating systems used rules combined with temporal linking systems trained on related tasks in order to anchor events to temporal expressions and entities to construct the timelines.

We propose a new approach to timeline extraction that uses the development data provided as distant supervision to generate noisy training data (Craven and Kumlien, 1999; Mintz et al., 2009a). More specifically, we heuristically align the target entity and the timestamps from the timelines with automatically recognized entities and temporal expressions in the documents. This noisy labeled data set allows us to learn models for the subtasks of anchoring events to temporal expressions and to entities, without requiring training models on additional data. Also, we improve the performance using joint inference for both anchoring subtasks. In our experiments, we show that our distantly supervised approach matches the state-of-the-art performance while joint inference further improves on it by 3.2 F-score points.

3.2 Timeline extraction

The task of timeline extraction given a target entity and a set of documents can be decomposed as follows. The initial stages are event mention extraction, target entity recognition, and temporal expression identification and resolution. The next stages are anchoring event mentions to target entities and temporal expressions. The final stages are event coreference resolution and ordering of the events in a timeline, which rely largely on their anchoring to temporal expressions. The TimeLine shared task had two tracks, A and B, the only difference being that in Track B the event mentions are provided in the input. We consider this track in this section and focus on learning the anchoring of events to temporal expressions and entities.

The development data provided in the context of the shared task consisted of documents related to *Apple* and gold timelines for six target entities. Evaluation was performed by extracting timelines from three document sets, each related to *Airbus*, *GM* and *Stock market* respectively. We used the official evaluation which is based on the metric introduced by UzZaman and Allen (2011) which assesses a predicted timeline versus the gold standard one using precision, recall and F-score over binary temporal relations between the events.

[6] SemEval-2015 Task 4: TimeLine (Cross-Document Event Ordering): http://alt.qcri.org/semeval2015/task4/

Documents:

DocId: 16844, DCT: 2010-06-08, Sentence: 2,3,4,5: *Yesterday*[2010-06-07] *, at this year 's Apple Worldwide Developers Conference (WWDC) , company CEO Steve Jobs*[Steve Jobs] *unveiled iPhone 4*[iPhone 4] *, along with the new iOS 4 operating system for Apple mobile devices . The announcement was long-awaited but not a very big surprise . In April , the technology blog Gizmodo obtained a prototype of the new phone*[iPhone 4] *and published details of it*[iPhone 4] *online . While introducing iPhone 4*[iPhone 4] *, at the annual conference , Jobs*[Steve Jobs] *started by hinting at the incident , saying , " Stop me if you 've already seen this .*

DocId: 17036, DCT: 2010-07-17, Sentence: 6,15: *Rather than recall the devices or offer a hardware fix , Jobs*[Steve Jobs] *said yesterday*[2010-07-16] *that Apple will offer a free case to anyone who has purchased an iPhone 4*[iPhone 4] *. [...] However , Jobs*[Steve Jobs] *admitted that the percentage of calls dropped on the iPhone 4*[iPhone 4] *was slightly greater than the percentage of calls dropped on the 3GS*[iPhone 3GS] *.*

DocId: 16900, DCT: 2010-06-16, Sentence: 6: *The newest iPhone*[iPhone 4] *, iPhone 4*[iPhone 4] *was introduced by Apple CEO*[Steve Jobs] *Steve Jobs*[Steve Jobs] *at the company 's 2010 Worldwide Developer 's Conference less than two weeks ago*[2010-06] *.*

DocId 16983, 2010-10-23, Sentence 10: *In his*[Steve Jobs] *keynote address Wednesday*[2010-10-20] *, Jobs*[Steve Jobs] *announced the release of Apple 's iLife '11 software suite , which includes the iPhoto , iMovie , and GarageBand programs .*

Timeline: *Steve Jobs*

1	2010-06-07	16844-2-unveiled	
1	2010-06-07	16844-5-introducing	16900-11-introduced
1	2010-06-07	16844-5-hinting	
1	2010-06-07	16844-5-saying	
2	2010-07-16	17036-6-said	
2	2010-07-16	17036-15-admitted	
3	2010-10-20	16983-10-address	
3	2010-10-20	16983-10-announced	

Figure 5: Example timeline for target entity *Steve Jobs*. The input to the system is the documents annotated with event mentions annotations and their Document Creation Time (DCT). The event mentions appearing in the timeline are identified by their document id-sentence index. The annotations for the target entities and temporal expression mentions need to be done by the system.

3.3 Distant supervision

In order to generate training data for anchoring event mentions to target entities and temporal expressions via distant supervision, we first need to identify them. For entity recognition we use approximate string matching combined with the Stanford Coreference Resolution System (Lee et al., 2013). For temporal expression identification and resolution to absolute timestamps we use the UWTime temporal parser (Lee et al., 2014).

Next we generate labeled instances as follows. For anchoring events to entities, we consider for each event mention the correct entity mention to be the nearest mention of the target entity in the same sentence, and all others to be incorrect. Similarly, for anchoring events to timestamps,

Table 9: Features to encode dependencies between events and target entities

Features	type
Measure distance in tokens between event and target entity mentions	local
Syntactic dependencies between event and target entity mentions (extracted from training corpus)	local
Check if subsequent events have the same stem and are attributed to the same target entity	global
Check if subsequent events are in the same sentence and are attributed to the same target entity	global
Check if subsequent events are both communication events and are attributed to the same target entity	global

we consider for each event mention the correct temporal expression to be the nearest temporal expression that exactly matches the timestamp according to the timeline (but not necessarily in the same sentence), and all others to be incorrect. The datasets generated will be noisy since correct anchors may be entity mentions and temporal expressions that are not the nearest ones. Further noise is expected due to errors in the entity recognition and temporal expression identification and resolution stages.

3.4 Event anchoring

After generating training data for anchoring event mentions to target entities and to temporal expressions with distant supervision, we now proceed to developing linear models for each of these tasks.

3.4.1 Classification

Using distant supervision we obtained examples of correct and incorrect anchoring of event mentions to entities and temporal expressions. Thus we learn for each of the two tasks a binary linear classifier of the form:

$$score(x, y, \mathbf{w}) = \mathbf{w} \cdot \phi(x, y) \tag{4}$$

where x is an event mention, y is the anchor (either the target entity or the temporal expression) and $\mathbf{w}$ are the parameters to be learned. The features extracted by ϕ represent various distance measures and syntactic dependencies between the event mention and the anchor obtained using Stanford CoreNLP (Manning et al., 2014). The temporal expression anchoring model also uses a few feature templates that depend on the timestamp of the temporal expression. The full list of features extracted by ϕ are denoted as local in Tables 9 and 10.

3.4.2 Alignment

The classification approach described is limited to anchoring each event mention to an entity or a temporal expression in isolation. However it would be preferable to infer the decisions for each task jointly at the document level and take into account the dependencies in anchoring different events, e.g. that consecutive events in text are likely to be anchored to the same entity, as shown in Figure 6, or to the same temporal expression. Capturing such dependencies can be crucial when the

Table 10: Features to encode dependencies between events and temporal expressions

Features	**type**
Measure distance in sentences between event mention and temporal expression	local
Measure distance in tokens between event mention and temporal expression	local
Syntactic dependencies between event mention and temporal expression (extracted from training corpus)	local
Check if temporal expression is before of after the event mention	local
Check if timestamp is in the future wrt the DCT	local
Check if timestamp is undefined (i.e. XX-XX-XXXX)	local
Check if timestamp is incomplete	local
Check if subsequent events and are linked to the same temporal expression	global
Check if subsequent events have the same stem and are linked to the same temporal expression	global
Check if subsequent events are in the same sentence and are linked to the same temporal expression	global
Check if subsequent events are communication events and are linked to the same temporal expression	global

correct anchor is not explicitly signalled in the text but can be inferred considering other relations and/or their ordering in text (Derczynski, 2013).

Defining our joint model formally, let $\mathbf{x}$ be a vector containing all event mentions in a document and $\mathbf{y}$ be the vector of all anchors (target entity mentions or temporal expressions) in the same document. The order of the events in $\mathbf{x}$ is as they appear in the document. Let $\mathbf{z}$ be a vector of the same length as $\mathbf{x}$ that defines the alignment between $\mathbf{x}$ and $\mathbf{y}$ by containing pointers to elements in $\mathbf{y}$, thus allowing for multiple events to share the same anchor. The scoring function is defined as

$$score(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) \tag{5}$$

where the global feature function Φ, in addition to the features returned by the local scoring function (Eq. 4), also returns features taking into account anchoring predictions across the document. Apart from features encoding subsequences of anchoring predictions, it also makes possible to make them dependent on the events, e.g. a binary indicator encoding whether two consecutive events with the same stem share the same anchor or not. The full list of local and global features extracted by Φ are presented in Tables 9 and 10. Predicting with the scoring function in Eq.5 amounts to finding the anchoring sequence vector $\mathbf{z}$ that maximizes it. To be able to perform exact inference efficiently, we impose a first order Markov assumption and use the Viterbi algorithm (Viterbi, 1967). Similar approaches have been successful in word alignment for machine translation (Blunsom and Cohn, 2006).

3.4.3 Post-processing

During testing, we need to construct the timeline for each target entity using the events that were predicted to be anchored to it and the timestamps of the temporal expressions each event was anchored to. Thus, we need to perform two additional tasks, event coreference and ordering. For the former we define a simple heuristic by which if two mentions have the same stems and timestamps then they refer to the same event. The only exception is that if two mentions represent

Documents:

DocId: 17036, DCT: 2010-07-17, Sentence: 9,10,11: *Jobs*[Steve Jobs] *also said that those who had already purchased a bumper will receive a full refund for the accessory. For consumers still dissatisfied with iPhone 4*[iPhone 4]*,* *Jobs*[Steve Jobs] *said that the phones can be returned for a refund as well.* *Jobs*[Steve Jobs] *acknowledged that "a very small percentage of users" were experiencing antenna issues, but dismissed the existence of an "Antennagate," saying that similar problems plague all cellular phones and that the iPhone*[iPhone 4] *issue "has been blown so out of proportion that it is incredible."*

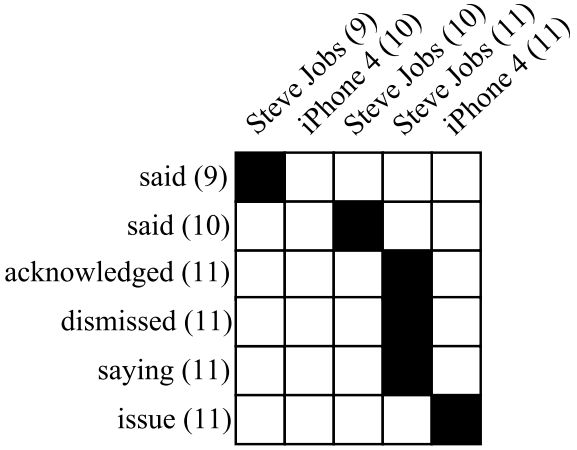

Figure 6: The correct alignment of events and target entity mentions is shown with the numbers in brackets denoting the index of the sentence in which the mention is found. The consecutive events acknowledged, dismissed and saying are anchored to entity Steve Jobs that was only mentioned once in the beginning of the sentence.

communication events (*said*, *announced* etc.), then they are resolved to different events when in the same document. We finally order the events according to their timestamp.

3.5 Results

We evaluate our system using the setup provided by the TimeLine task ensuring that the training and validation are performed only using the development data i.e. the *Apple* collection. All linear models were trained with the perceptron update rule (Pedregosa et al., 2011). We tuned the number of perceptron iterations by performing cross-validation using the development data by holding out the timeline for one target entity and training on the timelines for the remaining ones.

In Table 11 we compare the binary classification model (Our_System_Binary) against the alignment model (Our_System_Alignment) and show that the latter outperforms the former by a margin of 3.2 points in F-score, achieving a micro F_1-score of 28.58 across the three test corpora, thus confirming the benefits of joint inference. The only corpus in which joint inference did not help was *Stock* which has on average shorter event chains per document (Minard et al., 2015) and thus renders joint anchoring less likely to be useful.

We now compare our approach to the two participants in the TimeLine shared task with two runs each. The best-performing GPLSIUA team (Navarro and Saquete, 2015) used the TIPSem tool developed by Llorens et al. (2010) for temporal relation processing which extracts events and temporal expressions and uses a Conditional Random Field model to anchor them against each other.

Table 11: Results for our system and other participants in the SemEval 2015 Task 4: TimeLine.

	Airbus	GM	Stock	Total		
System	F_1	F_1	F_1	P	R	F_1
GPLSIUA_1	22.35	19.28	33.59	21.73	**30.46**	25.36
GPLSIUA_2	20.47	16.17	29.90	20.08	26.00	22.66
HeidelToul_1	19.62	7.25	20.37	20.11	14.76	17.03
HeidelToul_2	16.50	10.82	25.89	13.58	28.23	18.34
Our_System_Binary	17.99	20.97	**34.95**	25.97	24.79	25.37
Our_System_Alignment	**25.65**	**26.64**	32.35	**29.05**	28.12	**28.58**

However, TIPSem only considers anchoring of events to temporal expressions that are in the same sentence. GPLSIUA also used the semantic role labeler from SENNA (Collobert et al., 2011a) and OpenNER and anchored entities to events using a rule-based approach. The HeidelToul team (Moulahi et al., 2015) used HeidelTime Strötgen et al. (2013) to identify and resolve temporal expressions and developed a target entity mention identification tool similar to ours using Stanford CoreNLP (Manning et al., 2014). However, they rely on a rule-based approach for event anchoring. Our binary model matches the performance of the best system, and our alignment model exceeds it by 3.2 F_1-score points across, even though we do not use any off-the-shelf components developed for temporal relation extraction. Instead we rely on training data generated with distant supervision, and UWTime for temporal expression identification and resolution, for which the participants also used similar components.

3.6 Related work

In recent work, Laparra et al. (2015) also considered anchoring at the document-level in the context of the Track A of the TimeLine shared task, however they developed a rule-based approach. The structure features used in our joint inference approach encode similar intuitions, but we are learning model weights using distant supervision so that we can combine them more flexibly. And even though the noise in the trainng data generated with distant supervision is a concern, manual annotation of temporal relations is known to have low inter-annotator agreement rates[7] and thus also likely to be noisy.

Prior to the TimeLine shared task, TempEval (Verhagen et al., 2007) was the original task that focused on categorising the relations between events, temporal expressions and Document Creation Time using the the TimeML annotation language. The task classified only the relations between mentions in the same or consecutive sentences. The two following tasks, TempEval-2 (Verhagen et al., 2010) and TempEval-3 (UzZaman et al., 2013), added tasks for event and temporal expression identification as well as an end-to-end temporal relation processing task that was performed on raw text.

Beyond TempEval, McClosky and Manning (2012) used distant supervision in order to learn how to extract the temporal bounds for events in the context of the TAC temporal knowledge base population task (Ji et al., 2011). However they focus on learning real-world event ordering constraints (e.g. people go to school before university) instead of how events are reported in text.

[7] http://www.timeml.org/timebank/documentation-1.2.html

3.7 Conclusions

In this section we described a timeline extraction approach in which we generate noisy training data for anchoring events to entities and temporal expressions using distant supervision. By learning a binary classifier we match the state-of-the-art F_1-score for the Track B of the TimeLine shared task. We further improve this result by 3.2 F_1-score points using joint inference.

In future work we will focus on identifying event mentions instead of assuming them given as part of the input, so that our approach can be run on any text. Furthermore, we will explore how to evaluate it in the context of SUMMA in collaboration with BBC Monitoring Research.

3.8 Publications

Savelie Cornegruta and Andreas Vlachos. Timeline extraction using distant supervision and joint inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1936–1942, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1200

4 Convolutional 2D knowledge graph embeddings

4.1 Introduction

Knowledge graphs are databases of triples, where facts are represented in the form of relationships (edges) between entities (nodes). While knowledge graphs have important applications in search, analytics, recommendation, and data integration they suffer from incompleteness, that is, missing links in the graph. For example, in Freebase and DBpedia more than 66% of the person entries are missing a birthplace (Dong et al., 2014; Krompaß et al., 2015). Link prediction is the task to predict missing links in such a graph.

One focus of previous link prediction work lies in the number of model parameters. Knowledge graphs can be large and as such link predictors should scale in a manageable way with respect to both the number of parameters and computational costs to be applicable in real-world scenarios. It has also been shown that more complex models suffer from over-parameterisation and are thus prone to overfitting (Nickel et al., 2016a). Due to these limitations, link prediction models are often composed of simple operations, like dot products and matrix multiplications, over an embedding space and use a limited number of parameters.

In computer vision, convolutional layers are used to overcome these problems by using parameter sharing to learn multiple layers of non-linear features which are increasingly abstract and as such more expressive than shallower models (Krizhevsky et al., 2012).

Due to a range of regularisation techniques (Ioffe and Szegedy, 2015; Srivastava et al., 2014; Szegedy et al., 2016), convolutional architectures have little or no problems with overfitting even for networks with a depth close to a hundreds layers (Srivastava et al., 2015; He et al., 2016). Convolutional layers are highly parameter efficient, for example convolutional architectures have about 90% of their parameters in the fully connected hidden layers which compose features, while having only 10% of the parameters in convolutional layers which extract features (Krizhevsky et al., 2012). As such, convolutional layers constitute highly parameter efficient feature extractors which can model rich non-linear interactions and generalise effectively.

In this work we propose a neural link predictor, ConvE, that uses 2D convolution over embeddings to predict new links in knowledge graphs. While the bulk of our model's parameters is still in the relation and entity embeddings like in other link predictors, by using convolution as a weight sharing mechanism, we only use an additional 72 parameters to extract an extra layer of non-linear features which are then projected back to embedding space for scoring. We thus have a highly parameter efficient, scalable architecture, which generalises well and uses few additional parameters compared to other commonly used link prediction models. We make the following contributions:

- We introduce ConvE, a simple model that uses 2D convolutions over embeddings and achieves state-of-the-art performance on the WN18 (Bordes et al., 2013), FB15k (Bordes et al., 2013), YAGO3-10 (Mahdisoltani et al., 2015) and Countries (Bouchard et al., 2015) datasets.
- Using a 1-N approach for link prediction, where we predict scores for all possible links simultaneously, we speed up evaluation by several orders of magnitude. While useful for our model to get more predictions from each expensive convolution operation, this approach can also help speed up other link prediction models.

- Previous work by Toutanova and Chen (2015) noted that FB15k and WN18 contain many redundant, reversible relations, but they did not investigate the severity of this problem. We demonstrate the severity by designing a simple model which is based on a reversal rule that achieves state-of-the-art results on WN18 and FB15k, suggesting that successful models might learn this rule rather than to model the knowledge graph itself. We propose a new version of WN18, which follows the construction procedure of FB15k-237 to alleviate this problem.
- We show that the performance of our model compared to a shallow model, DistMult, increases when the network and test set involves nodes with high indegree or high PageRank. In particular, we show that the performance of our model increases relative to DistMult with increasing mean PageRank of nodes in the test set ($r = 0.83$). This suggests that our model is better at modelling nodes with high indegree.

4.2 Related work

Several neural link prediction models have been proposed in the literature, such as the Translating Embeddings model (TransE: Bordes et al., 2013), the Bilinear Diagonal model (DistMult: Yang et al., 2015) and its extension in the complex space (ComplEx: Trouillon et al., 2016) – we refer to Nickel et al. (2016a) for a recent survey on such models. The neural link prediction model that is most closely related to this work is most likely the Holographic Embeddings model (HolE: Nickel et al., 2016b), which uses cross-correlation – the inverse of circular convolution – for matching entity embeddings; it is inspired by holographic models of associative memory. However, HolE does not learn multiple layers of non-linear features and is thus theoretically less expressive than our model.

To the best of our knowledge, our model is the first model to use 2D convolutional layers when defining neural link prediction models for link prediction. *Graph Convolutional Networks* (GCN: Duvenaud et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2016) are a related line of research, where the convolution operator is generalised to use locality information in graphs. However, the GCN framework is limited to undirected graphs while knowledge graphs are naturally directed, and suffers from potentially prohibitive memory requirements (Kipf and Welling, 2016). Relational GCNs (R-GCN: Schlichtkrull et al., 2017) are a generalisation of GCNs developed for dealing with highly multi-relational data such as knowledge graphs – we include them in our experimental evaluations.

Several CNN-based models have been proposed in natural language processing (NLP) for solving a variety of tasks, including semantic parsing (Yih et al., 2011), sentence classification (Kim, 2014), search query retrieval (Shen et al., 2014), sentence modelling (Kalchbrenner et al., 2014), as well as traditional NLP tasks (Collobert et al., 2011b). However, most work in NLP uses 1D-convolutions, that is convolutions which operate over a temporal sequence of embeddings, for example a sequence of words in embedding space. In this work, we use 2D-convolutions which operate on a spatial level directly on embeddings. As we show later in section 4.8, this induces pixel-level spatial structure in our embeddings.

Using 2D convolutions has one major advantage for interactions between embeddings: Consider for example the case where we concatenate two rows of 1D embeddings — that is lining them up — a 1D convolution will be able to extract features of the interaction between these two embeddings at the concatenation point; if we concatenate two rows of 2D embeddings — that is stacking them

Table 12: Scoring functions $\psi_r(\mathbf{e}_s, \mathbf{e}_o)$ from neural link predictors in the literature, their relation-dependent parameters and space complexity.

Model	Score $\psi_r(\mathbf{e}_s, \mathbf{e}_o)$	Relation parameters	Space complexity
RESCAL (Nickel et al., 2011)	$\mathbf{e}_s^T \mathbf{W}_r \mathbf{e}_o$	$\mathbf{W} \in \mathbb{R}^{k \times k}$	$O(n_e k + n_r k^2)$
SE (Bordes et al., 2014)	$\left\lVert \mathbf{W}_r^L \mathbf{e}_s - \mathbf{W}_r^R \mathbf{e}_o \right\rVert_p$	$\mathbf{W}_r^L, \mathbf{W}_r^R \in \mathbb{R}^{k \times k}$	$O(n_e k + n_r k^2)$
TransE (Bordes et al., 2013)	$\lVert \mathbf{e}_s + \mathbf{r}_r - \mathbf{e}_o \rVert_p$	$\mathbf{r}_r \in \mathbb{R}^k$	$O(n_e k + n_r k)$
DistMult (Yang et al., 2015)	$\langle \mathbf{e}_s, \mathbf{r}_r, \mathbf{e}_o \rangle$	$\mathbf{r}_r \in \mathbb{R}^k$	$O(n_e k + n_r k)$
ComplEx (Trouillon et al., 2016)	$\langle \mathbf{e}_s, \mathbf{r}_r, \mathbf{e}_o \rangle$	$\mathbf{r}_r \in \mathbb{C}^k$	$O(n_e k + n_r k)$
ConvE	$f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega\right)\right)\mathbf{W}\right)\mathbf{e}_o$	$\mathbf{r}_r \in \mathbb{R}^k$, $\omega \in \mathbb{R}^{m \times n}$, $\mathbf{W} \in \mathbb{R}^{k \times k'}$	$O(n_e k + n_r k')$

— a 2D convolution will be able to extract features of interactions over the entire concatenation line. Thus 2D convolution is able to extract more features of interactions between two embeddings compared to 1D convolution.

4.3 Background

A *knowledge graph* $\mathcal{G} \triangleq \{(s, r, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ can be formalised as a set of triples (facts), each consisting of a relationship $r \in \mathcal{R}$ and two entities $s, o \in \mathcal{E}$, referred to as the *subject* and *object* of the triple. Each triple (s, r, o) denotes a relationship of type r between the entities s and o.

The *link prediction* problem can be formalised as a pointwise learning to rank problem, where the objective is learning a scoring function $\psi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$. Given an input triple $x = (s, r, o)$, its score $\psi(x) \in \mathbb{R}$ is proportional to the likelihood that the fact encoded by x is true.

In our case, the score of a relationships is defined as a deep convolutional network (LeCun et al., 1998).

Neural link predictors

Neural link prediction models (Nickel et al., 2016a) can be seen as multi-layer neural networks composed by an *encoding component* and a *scoring component*. Given an input triple (s, r, o), the encoding component maps entities $s, o \in \mathcal{E}$ to their distributed embedding representations $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$. In the scoring component, the two entity embeddings $\mathbf{e}_s$ and $\mathbf{e}_o$ are scored by a function ψ_r. The score of (s, r, o) is defined as $\psi(s, r, o) \triangleq \psi_r(\mathbf{e}_s, \mathbf{e}_o) \in \mathbb{R}$.

In Table 12 we summarise the scoring function of some link prediction models from the literature. The vectors $\mathbf{e}_s$ and $\mathbf{e}_o$ denote the subject and object embedding, where $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k$ in ComplEx and $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$ in all other models, and $\langle x, y, z \rangle \triangleq \sum_i x_i y_i z_i$ denotes the tri-linear dot product; $*$ denotes the convolution operator; f denotes a non-linear function.

4.4 Convolutional 2D embeddings of knowledge graphs

In this work we propose a neural link prediction model where the interactions between input entities and relationships is modelled by fully-connected and convolutional layers. Our model's main

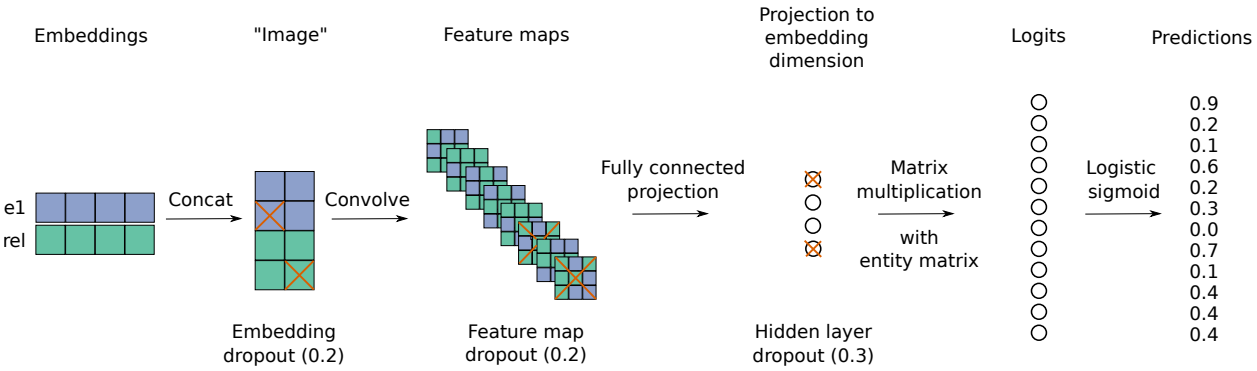

Figure 7: In the ConvE model the entity and relation embeddings are first reshaped and concatenated (steps 1, 2) and the resulting matrix is used as an input to a convolutional layer (step 3) the resulting feature map tensor is vectorised and projected in a k-dimensional space (step 4) and matched with all candidate object embeddings (step 5).

feature is convolution over 2D shaped embeddings. The architecture is summarised in Figure 7. Formally, the scoring function is defined as follows:

$$\psi_r(\mathbf{e}_s, \mathbf{e}_o) \triangleq \quad f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}_r}] * \omega\right)\right)\mathbf{W}\right)\mathbf{e}_o, \tag{6}$$

where f denotes a non-linear function, and $\overline{\mathbf{e}_s}$ and $\overline{\mathbf{e}_o}$ denote a 2D reshaping of $\mathbf{e}_s$ and $\mathbf{e}_o$, respectively: if $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, then $\overline{\mathbf{e}_s}, \overline{\mathbf{e}_o} \in \mathbb{R}^{k_w \times k_h}$, where $k = k_w k_h$.

In the feed-forward pass, the model performs a row-vector look-up operation in two embedding matrices, one for entities, denoted $\mathbf{E}^{|\mathcal{E}| \times k'}$ and one for relations, denoted $\mathbf{R}^{|\mathcal{R}| \times k'}$, where k and k' are respectively the entity and relation embedding dimensions, and $|\mathcal{E}|$ and $|\mathcal{R}|$ denote the number of entities and relations. The model then concatenates $\overline{\mathbf{e}_s}$ and $\overline{\mathbf{e}_o}$, and uses it as an input for a 2D convolutional layer with filters ω. Such a layer returns a feature map tensor $\mathcal{T} \in \mathbb{R}^{c \times m \times n}$, where c is the number of 2D filters, and m and n are the dimensions of the extracted feature maps. The tensor $\mathcal{T}$ is then reshaped in a $\mathrm{vec}(\mathcal{T}) \in \mathbb{R}^{cmn}$ vector, which is then projected in a k-dimensional space by a linear transformation parametrised by the matrix $\mathbf{W} \in \mathbb{R}^{cmn \times k}$ and matched with the object embedding $\mathbf{e}_o$ via a dot product. The convolutional filters and the matrix $\mathbf{W}$ are shared parameters, independent of the input entities s and o and the relationship r.

For training the model parameters, we apply a logistic sigmoid to the logits of the scores of (s, r, o), and minimise the following binary cross-entropy loss:

$$\mathcal{L}(p, t) = -\frac{1}{N}\sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)), \tag{7}$$

where p is the prediction and t the label.

We use rectified linear units as the non-linearity f for faster training (Krizhevsky et al., 2012), and batch normalisation after each layer to stabilise, regularise and increase rate of training convergence (Ioffe and Szegedy, 2015). We regularise our model by using dropout (Srivastava et al., 2014) in several stages: We dropout the embeddings, feature maps after the convolution operation and hidden units after the fully connected layer. We use Adam as optimiser (Kingma and Ba, 2015), and label smoothing to lessen overfitting due to saturation of output non-linearities at the labels (Szegedy et al., 2016).

4.4.1 Fast evaluation for link prediction tasks

Unlike other link prediction models which take an entity pair and a relation as a triple (s, r, o), and score it (1-1 approach), we take one (s, r) pair and score it against all entities $o \in \mathcal{E}$ simultaneously (1-N approach). In our architecture convolution consumes about 75-90% of the total computation time of the model and thus it is important to minimise the number of convolution operations. If we take a naive 1-1 approach, then a training pass and an evaluation with a convolution model on FB15k – one of the dataset used in the experiments – takes 2.4 minutes and 3.34 hours, respectively, using a high-end GPU with a batch size of 128 and embedding size 128. Using a 1-N approach the respective numbers are 45 and 35 seconds – a considerable improvement of over 300x in terms of evaluation time. Usually, in a 1-1 approach, the batch size is increased to speed up evaluation (Bordes et al., 2013), but this is not feasible for convolutional models since the GPU memory requirements quickly blows up for large batch sizes when one uses convolution. Thus the 1-N evaluation is of practical importance, since our model would be computationally expensive to evaluate otherwise.

Do note that any 1-1 model can make use of 1-N evaluation. Thus this practical trick in speeding up the evaluation is applicable for all standard link prediction models which usually operate using a 1-1 approach.

4.5 Experiments

4.5.1 Knowledge graph datasets

For evaluating our proposed model, we use a selection of common knowledge graphs from the literature. WN18 (Bordes et al., 2013) is a subset of WordNet which consists of 18 relations and 40,943 entities. Most of the 151,442 triples have hyponym and hypernym relations and thus WN18 is marked by relations arranged in a strictly hierarchical structure. FB15k (Bordes et al., 2013) is a subset of Freebase which contains about 15k entities with 1,345 different relations. A large fraction of content in this knowledge graph deals with movies/actors/awards and sports/sport teams. YAGO3-10 (Mahdisoltani et al., 2015) is a subset of YAGO3 which consists of entities which have a minimum of 10 relations each. It has 123,182 entities and 37 relations. Most of the triples deal with descriptive attributes of people (citizenship/gender/profession).

Countries (Bouchard et al., 2015) is a benchmark dataset that is useful to evaluate a model's ability to learn long-range dependencies between entities and relations. It consists of three sub-tasks which increase in difficulty in a step-wise fashion. It consists of three types of entities: countries (e.g. Germany, Italy), sub-regions (e.g. Western Europe, South America) and regions (e.g. Europe, Americas); and two relations: NeighborOf(country, country) and LocatedIn(country, sub-region/region). The task is to predict the region of a given country. It is evaluated in terms of area-under-the-curve for precision/recall (AUC-PR). The task has three levels of difficulty, S1, S2, S3, where each step increases the path-length or indirection which one needs to traverse the dataset graph in order to find a solution. In S1 all regions of test-case countries are removed from the training set (solution: Country → sub-region → region); in S2 one also removes the sub-regions (solution: Country → neighbour → region); in S3 one also removes the region of all the neighbors (solution: Country → neighbour → sub-region → region).

We also introduce a new dataset, WN18RR, which is an alteration of WN18 which aims to address the shortcoming of WN18: It was first noted by Toutanova and Chen (2015) that the test sets of

WN18 and FB15k contain mostly reversed triples that are present in the training set, for example the test set contains $(s, \text{hyponym}, o)$ while the training set contains the reverse $(o, \text{hypernym}, s)$. Toutanova and Chen (2015) introduced FB15k-237, a subset of FB15k where reversing relations are removed, to create a dataset without this property. However, they did not explicitly demonstrate the severity of this problem, which might explain the fact that further research kept using these datasets for evaluation without addressing this issue.

In Section 4.5.2 we introduce a simple reversal rule which demonstrates the severity of this bias by achieving state-of-the-art results on both WN18 and FB15k. One might argue, that a good relational model should learn how to reverse relations in addition to more complex aspects of the dataset, but if this is our evaluation goal we should design more controlled datasets and experiments where it is clear what a relational model learns. By creating WN18RR, we seek to reclaim WN18 as a dataset which tests a models ability to model a general knowledge graph which cannot easily be completed using a single rule. As such, we do not recommend the usage of FB15k and WN18 in further research. Instead, we recommend the usage of FB15k-237, WN18RR, and YAGO3-10 that do not suffer from these issues.

4.5.2 Experimental setup

We selected the hyperparameters of our ConvE model via grid search according to the mean reciprocal rank (MRR) on the validation set. Hyperparameter ranges for the grid search were the following – embedding dropout in $\{0.0, 0.1, 0.2\}$, feature map dropout in $\{0.0, 0.1, 0.2, 0.3\}$, projection layer dropout in $\{0.0, 0.1, 0.3, 0.5\}$, embedding size in $\{100, 200\}$, batch size in $\{64, 128, 256\}$, learning rate in $\{0.001, 0.003\}$, label smoothing in $\{0.0, 0.1, 0.2, 0.3\}$. Besides the grid search, we also tried modifications of the 2D convolution layer in our models: We tried replacing it with fully connected layers, and 1D convolution, but these performed consistently worse and we abandoned them. We also experimented with different filter sizes and found that we only receive good results if the first convolutional layer uses small filters, that is 3x3 filters. We found that the following combination of parameters works well on WN18, YAGO3-10 and FB15k: embedding dropout 0.2, feature map dropout 0.2, projection layer dropout 0.3, embedding size 200, batch size 128, learning rate 0.001, label smoothing 0.1. For the Countries dataset, we increase embedding dropout to 0.3, hidden dropout to 0.5 and set label smoothing to 0.

We use early stopping using the mean reciprocal rank (WN18, FB15k, YAGO3-10) and AUC-PR (Countries) statistics on the validation set which we evaluate every three epochs. Unlike the other datasets, for Countries the results have a high variance, as such we average 10 runs and produce 95% confidence intervals.

Baseline – reversal model: WN18 and FB15k has been noted to contain many reversible relations (Toutanova and Chen, 2015), for instance, a test triple (feline, *hyponym*, cat) can easily be mapped to a training triple (cat, *hypernym*, feline): knowing that hyponym is the inverse of hypernym allows you to easily predict the vast majority of test triples.

For such a reason, to test the redundancy of the datasets, we also investigated a simple rule-based baseline model, which we will refer to as the *reverse model*. This model looks for relationships which are the reverse of each other, such as hypernym and hyponym. We extract these relationships automatically from the training set: given two relation pairs $r_1, r_2 \in \mathcal{R}$, we check whether (s, r_1, o) implies (o, r_2, s), and vice-versa. If the presence of (s, r_1, o) co-occurs with the presence of (o, r_2, s)

Table 13: Link prediction results on WN18 and FB15k

	WN18					**FB15k**				
			Hits					Hits		
	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult	902	0.822	0.936	0.914	0.728	97	0.654	0.824	0.733	0.546
ComplEx	–	0.941	0.947	0.936	0.936	–	0.692	0.840	0.759	0.599
Gaifman	**352**	–	0.939	–	0.761	75	–	0.842	–	0.692
ANALOGY	–	**0.942**	0.947	0.944	**0.939**	–	0.725	0.854	0.785	0.646
R-GCN	–	0.814	0.964	0.929	0.697	–	0.696	0.842	0.760	0.601
ConvE	504	**0.942**	0.955	0.947	0.935	**64**	0.745	**0.873**	**0.801**	0.670
ReverseModel	602	0.857	**0.969**	**0.958**	0.757	1563	**0.759**	0.786	0.771	**0.743**

Table 14: Link prediction results on WN18RR and FB15k-237

	WN18RR					**FB15k-237**				
			Hits					Hits		
	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult	**5110**	0.425	0.491	0.439	0.389	254	0.241	0.419	0.263	0.155
ComplEx	5261	**0.444**	**0.507**	**0.458**	**0.411**	**248**	0.240	0.419	0.263	0.152
R-GCN	–	–	–	–	–	–	0.248	0.417	0.258	0.153
ConvE	7323	0.342	0.411	0.360	0.306	330	**0.301**	**0.458**	**0.330**	**0.220**
ReverseModel	13417	0.360	0.360	0.360	0.360	7124	0.007	0.012	0.008	0.004

at least 99% of the time, we say that r_1 implies r_2 ($r_1 \implies r_2^-$) and we use this reversal rule for predicting test triples. We use the training set to check for reversal rules. At test time we check if the test triple has reversal matches outside the test set, if k matches are found we sample a permutation of the top k ranks for these matches; if no match is found we select a random rank for the test triple.

4.6 Results

Similarly to (Yang et al., 2015; Trouillon et al., 2016; Niepert, 2016), we here focus on reporting results in a filtered setting, that is we rank triples only against scores for all possible entity combinations of unknown triples and we do not rank against combinations of existing, known triples. Our results on the standard benchmarks FB15k and WN18 are shown in Table 13, results on the datasets with reversing relations removed are shown in Table 14; results on YAGO3-10 and Countries are shown in Table 15.

Strikingly, the reverse model baseline achieves state-of-the-art on many different metrics on both, FB15k and WN18 datasets. However, it fails to pick up on reversible relations on YAGO3-10 and FB15k-237. On WN18 our reverse model achieves a good score due to self-reversing relationships

Table 15: Link prediction results on YAGO3-10 and Countries

	YAGO3-10					Countries		
			Hits			AUC-PR		
	MR	MRR	@10	@3	@1	S1	S2	S3
DistMult	5926	0.337	0.540	0.379	0.237	**1.000±0.000**	0.721±0.122	0.516±0.070
ComplEx	6351	0.355	0.547	0.399	0.258	0.965±0.021	0.571±0.104	0.430±0.072
ConvE	**2792**	**0.523**	**0.658**	**0.564**	**0.448**	**1.000±0.000**	**0.985±0.013**	**0.856 ±0.051**
ReverseModel	60251	0.015	0.022	0.017	0.010	–	–	–

like ”similar to” which still exist in the dataset after using the procedure which is also used for the FB15k-237 dataset.

Our proposed model, ConvE, achieves state-of-the-art performance for all metrics on YAGO3-10, for some metrics on FB15k, and it does well on WN18. On Countries, it solves the S1 and S2 tasks, and does well on S3, scoring better than other models like DistMult and ComplEx

For FB15k-237, we could not replicate the basic model results from Toutanova et al. (2015), where the models in general have better performance than what we can achieve. Compared to Schlichtkrull et al. (2017) our results for standard models are a bit better then theirs and on-a-par with their R-GCN model.

4.7 Analysis

4.7.1 Looking at indegree and PageRank

Our main hypothesis for the good performance of our model on datasets like YAGO3-10 and FB15k-237 compared to WN18RR is that these dataset contain nodes with very high relation-specific indegree. For example the node ”United States” (entity embedding) with edges ”was born in” (relation embedding) has an indegree of over 10,000. Many of these 10,000 nodes will be very different from each other (actors, writers, politicians, business people) and our main hypothesis is that models that learn multiple layers of non-linear features like ours have an advantage over shallow models to capture all these constraints for such high indegree nodes.

However, for simpler datasets, like WN18 which mainly consists of hypernym/hyponym relations which often have an indegree of one (there is often only one generalisation for a given concept), we have nodes with small relation-specific indegree and thus a linear model might be sufficient, easier to optimise, and thus be able to find a better local minimum.

In this section we compare DistMult, a model that uses a simple tri-linear dot product, and our model, ConvE, that learn multiple layers of non-linear features to analyse the effect these high indegree nodes on performance.

We test our hypothesis in two ways. We remove triples which contain nodes which have a relation-specific indegree of greater than two for FB15k; and we remove triples which contain nodes which have a relation-specific indegree less than two for WN18. On these datasets we hypothesise that compared to DistMult, (1) our model will perform worse on FB15k (relatively more nodes with

low indegree), and (2) our model will perform better on WN18 (relatively more nodes with high indegree). Indeed, we find that both hypotheses hold: For (1) on FB15k we have ConvE 0.586 Hits@10 vs DistMult 0.728 Hits@10; for (2) on WN18 we have ConvE 0.952 Hits@10 vs DistMult 0.938 Hits@10. This shows that our model indeed might have an advantage when modelling nodes with high indegree.

To verify this hypothesis further we look at PageRank (Page et al., 1999), which is a measure of centrality of a node. PageRank can also be seen as a measure of the recursive indegree of a node, that is, the PageRank value of a node is proportional to the indegree of this node, its neighbours indegree, its neighbours-neighbours indegree and so forth scaled relative to all other nodes in the network.

In line with our argument above, we expect that nodes with high PageRank are more difficult to model, since one entity embedding needs to capture numerous constraints with other entity embeddings, and additionally, many of its neighbours, which by definition of high PageRank have often high indegree, will need to capture numerous additional constrains and so forth.

To test this hypothesis, we calculate the PageRank for each dataset as a measure of centrality. We find that the most central nodes in WN18 have a PageRank value more than one order of magnitude smaller than the most central nodes in YAGO3-10 and Countries, and about 4 times smaller than the most central nodes in FB15k. When we look at the mean PageRank of nodes contained in the test sets, we find that the difference of performance in terms of Hits@10 between DistMult and ConvE is roughly proportional to the mean test set PageRank, that is, the higher the mean PageRank of the test set nodes the better does ConvE compared to DistMult and vice-versa. See Table 16 for these statistics. The correlation between mean test set PageRank and relative error reduction of ConvE compared to DistMult is strong with $r = 0.83$. This gives additional evidence that our model has an advantage at modelling nodes with high (recursive) indegree.

To verify if this behaviour is caused by learning multiple layers of of non-linear features, we remove the convolutional layer from our model and replace it with either two fully connected layers or with a 1D convolutional layer while keeping the rest of the architecture consistent. We then train all three models on FB15k and compare the scores. If multiple layers of non-linear features would be the decisive factor we would expect similar score for all three models. We find that the fully connect architecture performs much worse then 1D or 2D convolution, with Hits@10 0.466 (MLP) vs 0.821 (1D) vs 0.873 (2D), respectively.

Experimentally, this shows that the multiple layers of non-linear features are not the decisive factors, but rather the ability of a layer to extract useful features. This is similar to the performance of multi-layer perceptrons compared to convolutional networks in computer vision — both models learn non-linear features, but convolutional layers can extract more relevant features which lead to better generalisation. The differences between 1D and 2D convolution suggests that features learned from the interaction of two embeddings (2D) are more powerful than just modelling the embeddings with convolution (1D).

In conclusion, we believe that the increased performance of our model compared to a standard link predictor, DistMult, can be partially be explained due to our model's ability to model nodes with high indegree with greater precision. We show that this behaviour might be due to the ability of 2D convolutional layers to learn features of interactions between entities and relation embeddings. This begs the question what kind of spatial structure 2D convolution is inducing in the embedding space.

Table 16: Mean PageRank $\times 10^{-3}$ of nodes in the test set vs reduction in error in terms of AUC-PR or Hits@10 of ConvE wrt. DistMult.

Dataset	PageRank	Error Reduction
WN18RR	0.104	0.91
WN18	0.125	1.28
FB15k	0.599	1.23
FB15RR	0.733	1.17
YAGO3-10	0.988	1.91
Countries S3	1.415	3.36
Countries S1	1.711	0
Countries S2	1.796	18.6

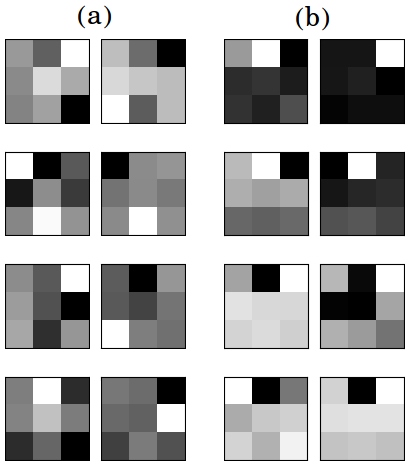

Figure 8: Visualisation of convolutional filters for (a) WN18 and (b) YAGO3-10.

4.8 Spatial structure of 2D embeddings

Here we test if the 2D embeddings contain spatial structure. We use Moran's I test which tests the null hypothesis that the global spatial autocorrelation is zero, or in other words, that the random variable does not have any global spatial structure (Moran, 1950). We use the PySal package (Rey and Anselin, 2010) to carry out the test.

We normalise each "image" to have mean zero and variance and apply the Moran's I test to up to 100 samples of entity and relation embeddings. We count the proportion of significant tests to get evidence if these embeddings have some structure in general. We find that Moran I tests on entity embeddings are significant near chance level, with less than 10% of tests being significant at the $p = 0.05$ level. For relation embeddings we find weak evidence for spatial structure for YAGO3-10 where 44% of relations are significant. These results suggest that 2D entity embeddings generally do not have global spatial structure and that only some of the relation embedding have global spatial structure.

However, visualisation of YAGO3-10 and WN18 convolution filters, as seen in Figure 8 suggest that these filters match contrasting "pixel" values, that is, the filters match patterns where one pixel or a group of pixels stands out relative to its neighbourhood. These pixel-feature filters give some hints why Moran's I test failed to pick up spatial patterns, since a spatial pattern in embeddings must be larger than a few contrasting pixels to be significantly different from noise. If the 2D convolutions would not pick up on any spatial features, we would expect that they lead to poor model performance so this means that these filters must work on pixel-level structures. This also explains why convolutions larger than 3x3 do not work well in the first convolutional layer since larger filters sum pixel-level features together with their surroundings thus diluting information in pixel-level structures.

4.9 Conclusion and future work

Here we introduced ConvE, a link prediction model that uses 2D convolution over embeddings and multiple layers of non-linear features to model knowledge graphs. This model uses few parameters

and is computationally efficient due to a 1-N approach of link prediction and thus scales well with increasing knowledge graph sizes. Our model achieves state-of-the-art results on several existing knowledge graph datasets. However, building on previous work, we also show that a simple reverse model for relations can achieve state-of-the-art results on WN18 and FB15k thus questioning if models on this dataset actually learn general link prediction rather than learning this reversal rule. We introduce WN18RR to address this issue for WN18 and we recommend using FB15k-237 over FB15k for future research.

In our analysis we show that the performance of our model compared to a common link predictor, DistMult, can partially be explained by its ability to model nodes with high (recursive) indegree. Tests for spatial autocorrelation reveal that the entity and relation embeddings in general do not have any significant spatial structure, but that 2D convolutions on embeddings instead learn structures of contrasting pixels.

Our model is still shallow compared to convolutional architecture found in computer vision and future work might deal with convolutional models of increasing depth. Further work might also look at the interpretation of 2D convolution or how to enforce large-scale structure in embedding space and thus make convolutional filters learn feature extractors for large scale structures in embedding interactions.

4.10 Publications

- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. arXiv e-print arXiv:1707.01476 [cs.LG], University College London, July 2017. URL https://arxiv.org/abs/1707.01476

5 One shot relation extraction with factorization machines

5.1 Introduction

Extracting the relations between entities of interest plays a useful role in many natural language understanding systems, including those used for various tasks such as question answering and automatic knowledge base population, resulting in several methods and techniques being used for this task (Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2006; Mintz et al., 2009b; Yao et al., 2011; Surdeanu et al., 2012). Extracting such information is key to the goals of the SUMMA project, and particularly relevant to the external media monitoring use case, in order to assist the users in understanding complex storylines involving new entities and relations.

Recently Riedel et al. (2013a) proposed a promising approach based on matrix factorization. This approach casts the problem of extracting relations between entities as one of link prediction over a universal schema consisting of the union of textual surface patterns, structured knowledge base relations and entities. In this framework, facts from the knowledge base stipulating that a certain relation holds among two entities provide supervision signal for learning relation extractors.

However, the need often arises in practice, for example in a constantly evolving news arena, to learn extractors for new relations for which there is limited training data, as is the case when a knowledge base needs to be extended to new relations for which there are only a few known facts. This learning setting, where a model is allowed to have one or few learning instances per class, has also been referred to as one-shot learning (Miller et al., 2000; Fei-Fei et al., 2006). Rocktäschel et al. (2015b) and Demeester et al. (2016) proposed to combine universal schema with propositional rules mined from WordNet to improve relation extraction learning with limited data.

Here, we describe the use of a Factorization Machine (FM) model (Rendle, 2010) for learning extractors for knowledge base relations for which there is limited existing training data. In the context of relation extraction, FMs can be seen as a geneneralization of matrix factorization that allows us to exploit information about the entities that is readily available within the text itself. While FM models have been explored for relation extraction by Petroni et al. (2015), their effectiveness was not investigated within the context of limited supervision data. Furthermore, unlike Petroni et al. (2015) we propose to model the correlations between knowledge base relations and surface patterns in the context of FMs. Following the aforementioned previous work, we evaluate our proposed approach on the *New York Times* dataset of Riedel et al. (2013a).

Our contributions are threefold: (i) we investigate the use of FMs for one-shot relation extraction and obtain improvements of 5% points in area under the weighted Mean Average Precision (wMAP) curve compared to matrix factorization, (ii) we demonstrate that by modelling the correlations between knowledge base relations and surface patterns we achieve performance equivalent to matrix factorization combined with propositional rules, despite not using such additional supervision in our approach, and (iii) when using the full training data our approach achieved a gain of 3% points in wMAP compared to the best matrix factorization-based result.

5.2 Background

5.2.1 One shot learning

The notion of one-shot learning, which has also been explored in computer vision (Miller et al., 2000; Fei-Fei et al., 2006), is used to describe the learning setting where the model is required to generalize from only one or few example instances per class. This is a realistic scenario when there are classes for which training data is limited, for instance when having to learn a classifier for a new type of object in computer vision or when a knowledge base needs to be extended to new relations for which limited learning examples are available. In contrast with the zero-shot learning (Larochelle et al., 2008) setting, where the model is not allowed any example labels, one-shot allows for one or few example labels per class. We consider the one-shot learning setting to be realistic, as limited supervision can often be easily obtained for new classes, for instance by asking the user to provide some examples for the new relation.

5.2.2 Relation extraction with universal schema

Universal Schema (Riedel et al., 2013a) is an approach to relation extraction that jointly embeds textual surface patterns, knowledge base relations and entities in a common embedding space through matrix factorization. It sidesteps the problem of aligning relations to sentences from the training corpus, which can lead to semantic drift in distantly supervised relation extraction approaches. It achieves this by performing joint inference across textual surface patterns and the entities and relations in a knowledge base. Rocktäschel et al. (2015b) and Demeester et al. (2016) inject prior knowledge in the form of logical rules to improve relation extraction learning for new relations with zero or few training labels. While their experiments were also carried out within the framework of universal schema-based relation extraction, they considered the use of propositional logic rules, which for instance, can be mined from external knowledge bases (which are often incomplete themselves) or obtained from ontologies such as Wordnet (which may not be readily available, especially for a new domain). We instead investigate the use of information which is available from the text itself, and which can be incorporated in FM models.

5.3 Model description

5.3.1 Factorization machines

We now describe factorization machines (FMs, Rendle (2010, 2012)) upon which we will develop our relation extraction approach. FMs were proposed in the context of recommender systems as a way to learn effective scoring functions with sparse inputs, in order to assess how likely is that a user-item combination occurs in reality. More concretely, they model the scoring of a possibly sparse, real-valued input feature vector $\boldsymbol{f} \in \Re^d$ according to the following equation:

$$s(\boldsymbol{f}) = \sum_{m=1}^{d} b_m f_m + \sum_{m=1}^{d} \sum_{n=m+1}^{d} \langle \phi_m, \phi_n \rangle f_m f_n \tag{8}$$

The first summand is a linear model, where each feature f_m is weighted by a corresponding feature weight $b_m \in \Re$. The second summand captures the interaction between all possible feature pairs under a low-rank assumption. Each feature f_m has a corresponding embedding $\phi_m \in \Re^k$ with $k <<$

d, and the interaction between two features is captured via their dot product $\langle \phi_m, \phi_n \rangle$ multiplied by the product of their values in the instance $f_m f_n$. The dot products among all feature pairs represent the weights that we would have in a model having a weight for each feature combination ($d(d-1)/2$ weights) but with fewer parameters (kd) and thus easier to learn from less and/or sparse data. Modeling feature interactions is crucial in FMs; e.g. in the context of recommender systems some features would represent the item and others the user, and the linear component of the model would only capture that some users tend to buy more items or that some items are more popular among users. Only the feature embeddings that are used to capture their interactions can inform us whether a particular user will buy a particular item. Note that the above equation represents an order-2 FM which captures interactions between pairs of features, but higher order FMs can capture interactions among feature groups of higher cardinality at additional computational cost.

An alternative view is that we learn the rank-k factorization of the matrix containing the weights for each feature pair, hence the name factorization machines. Rendle (2010) showed that a FM model is effective in several learning settings, even those with sparse features, and is also capable of approximating the behaviour of many matrix and tensor factorization models. In the following section we leverage its ability to learn feature interactions from sparse data to incorporate contextual information into our relation extraction approach and improve its accuracy.

5.3.2 Proposed model

We now describe how we apply FMs to learning relation extractors. Let $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{S}$ be the set of entity pairs, relations (knowledge base and surface ones) and textual surface patterns respectively. We represent a candidate fact as a triple (r, t, c^t) consisting of a relation $r \in \mathcal{R}$, an entity pair $t \in \mathcal{T}$ and the surface patterns forming the contextual neighbourhood c^t of the entity pair. We generate $\mathbf{f}$, its feature vector by concatenating vectors encoding each of these elements. The relation r and tuple t are encoded as one-hot feature vectors of dimensionalities $|\mathcal{R}|$ and $|\mathcal{T}|$ respectively. The contextual neighbourhood feature vector represents the counts of surface patterns that have been observed together with tuple t in a text corpus, normalized to sum to one. The intuition behind using the contextual neighbourhood features being that they provide evidence of the surface patterns that are descriptive of the entity pair in the text corpus and at the same time allowing the model to learn which surface patterns are indicative of knowledge base relations. Thus the model would be able to draw on statistical evidence from surface patterns across a text corpus in order to derive more reliable estimates for the interaction factors of relations. This also gives us the benefit of making the most of surface relations, which are easily obtained but noisy, to learn with very few annotation labels for relations. We can thus exploit any abundant text resource (the web, for instance) to learn relation extractors with very few supervision labels for a new relation.

For example, the first row in Figure 9 represents that the tuple *Paris,France* was observed with the surface relation *"is a city in"* and that the same tuple was observed with two surface patterns in its contextual neighborhood, *"is a city in"* and *"is a part of"*, hence each of them have a value of 0.5. Similarly, the sixth row represents that the same tuple *Paris,France* with the same contextual neighborhood having the KB relation *is_the_capital_of*. This allows the FM model to learn the interaction between the surface patterns *"is a city in"* and *"is a part of"* and the KB relation *is_the_capital_of*. Furthermore consider that we want to predict which is a more likely entity tuple between *London, United Kingdom* and *London, France* for the knowledge base relation *"is_located_in"*. Observe that the tuples *London, United Kingdom* and *Paris, France* have more neighbourhood context overlap than the tuple *London, France*. The proposed model would be

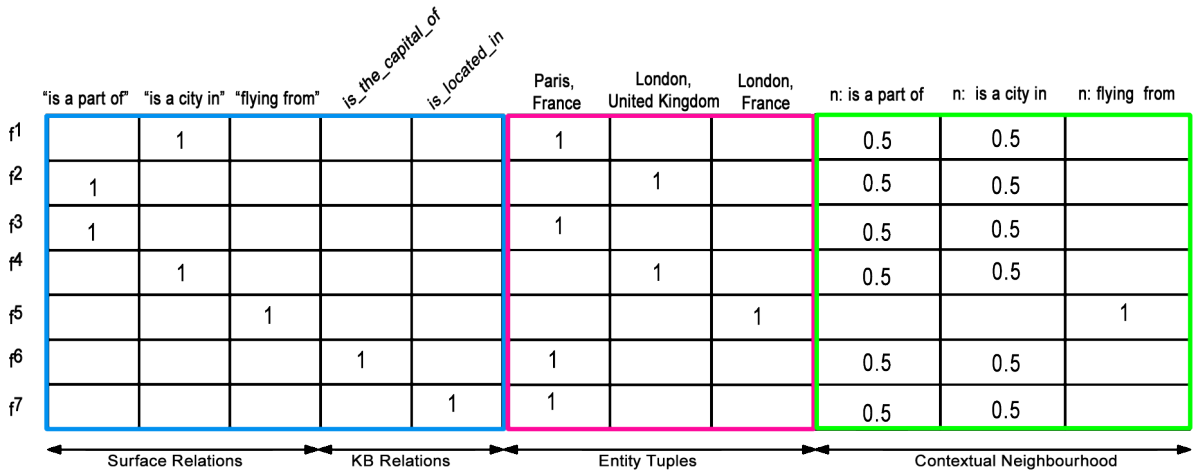

Figure 9: Input observations as a matrix with contextual neighbourhood information

aware of such correlations to give a higher score for the fact *(London,is_located_in,United Kingdom)* than *(London,is_located_in,France)*.

Since the feature vector $\mathbf{f}$ is very sparse as it consists of the one-hot encoded surface/KB relations and entity tuples and the contextual neighbourhood where most surface patters for a given tuple will have 0 value. We exploit this in order to accelerate the computation of Equation 1 for a candidate fact by ignoring the features with value of 0 and considering only the active ones A and their corresponding vector representations, which can be substituted into the FM model equation to arrive at the score for a fact:

$$s(\mathbf{f}) = \sum_{a \in A} b_a f_a + \sum_{a \in A, a' \in A \setminus a} \langle \phi_a \phi_{a'} \rangle f_a f_{a'} \tag{9}$$

Petroni et al. (2015) also proposed to formulate relation extraction learning with factorization machines using as additional features (i.e. features beyond the entities, the KB and the surface relations) article metadata, tuple type information and bag of words from the sentence of the extraction. However, the first two require additional preprocessing and/or human input, while the latter is only available for the rows representing facts mentioned in text, not facts from the KB, thus their use is limited. Instead, by adding the surface patterns from the contextual neighborhood as features we are able to capture the correlations between them and the KB relations that was not possible in the formulation of Petroni et al. (2015). In the left-hand part of the matrix in Figure 9 each instance has either a surface relation or a KB relation active, thus their correlations will be ignored unless we consider the context neighbourhood on the right-hand side.

5.3.3 Objective formulation

Given a text corpus, we aim to extract relations between entities of interest, with limited training data from the knowledge base and learn a model that can differentiate between true and false facts, i.e. assign high scores to the former and lower scores to the latter using equation 2. However, only examples of observed true relations between entities (positive facts) are available at training time. In order for the model to effectively discriminate between positive and negative facts, it needs to

have also seen examples of negative facts. One way to achieve this is to treat observed relations as true facts and all unobserved relations between entities as false facts. However since the facts we seek to extract are unobserved, this carries the risk that we treat plausible relations between entities as negative, which can consequently lead to inferior model performance. Following previous work (Riedel et al., 2013a; Petroni et al., 2015), we make use of an alternative approach, which is to instead treat unobserved facts as unknowns, and left for the model to infer. This is achieved using a ranking-based objective, which optimizes to rank observed facts higher that unobserved ones. Concretely, we make use of the Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) objective, which optimizes for the maximal difference between the score of observed and unobserved facts. Given a set of observed F^+ and unobserved F^- facts, we estimate model parameters Θ that satisfy the following objective:

$$\arg\min_{\Theta} -\sum_{\substack{\mathbf{f}^+ \in F^+ \\ \mathbf{f}^- \in F^-}} \log\left(1 + e^{\delta(\mathbf{f}^+, \mathbf{f}^-)}\right) + \lambda ||\Theta||^2 \tag{10}$$

where $\delta(\mathbf{f}^+, \mathbf{f}^-) = \mathbf{s}(\mathbf{f}^+) - \mathbf{s}(\mathbf{f}^-)$ and λ is a regularization hyper parameter. The objective (10) essentially maximizes the difference $\delta(\mathbf{f}^+, \mathbf{f}^-)$ between the scores of observed and unobserved facts.

Note that the set F^- is unobserved and is generated automatically from F^+ by random sampling. Specifically, in each iteration and for every positive fact $\mathbf{f}^+$ in the current batch, we fix the relation r and randomly select an entity pair $t' \in E$, such that the triple $(r, t', c^{t'})$ has not been observed.

5.4 Training and evaluation

For all experiments, we make use of a latent dimension size of 100, L_2 regularization penalty of 0.01, and ran our model for 1000 epochs. Our system is implemented in Tensorflow (Abadi et al., 2015), and uses Adam (Kingma and Ba, 2015) for optimization, with a learning rate of 1×10^{-4} and batch size of 1024. We sample one unobserved fact at random per positive fact during training.

We make use of the same evaluation setup as Riedel et al. (2013a), who retrieved for each relation the top 1000 entity tuples from each system, the top 100 of which is then pooled and manually annotated. These provided a set of results that is used to compute precision measures for each system. We computed Mean Average Precision (MAP) and weighted Mean Average Precision (wMAP) for each run. While MAP computes the expectation of average precision scores across all the relations for each system, weighted MAP takes into account the number of true facts for each relation in computing this expectation.

5.5 Experiments and results

For our experiments, we make use of the dataset of Riedel et al. (2013a), which consist of data from the New York Times (NYT) corpus (Sandhaus, 2008). The corpus has been preprocessed with a named entity recogniser and the entities have been linked, where possible, with their corresponding Freebase (Bollacker et al., 2008a) entities. The shortest dependency path between each pair of entities in a sentence has also been extracted as the textual surface relation. In our one-shot experiments, we perform evaluations with a fraction $\tau \in [0, 0.5]$ of the training labels for each relation. Note that we use the same dimensionality for the embeddings and the same pre-processing (named entity recognition and linking, syntactic parsing) as the approaches we are comparing against in order to ensure a fair comparison.

Table 17: Results using the full training dataset. The # column is the number of true facts in test pool. Winners are in bold, tied winners in italics.

Relation	#	M09	Y11	S12	R13-N	R13-F	R13-NF	FM	FM+n
person/company	104	0.66	0.63	0.69	0.72	0.75	0.75	0.79	**0.80**
location/containedby	72	0.46	0.42	0.51	0.41	**0.69**	0.67	0.68	0.68
person/nationality	27	0.14	**0.41**	0.13	0.14	0.20	0.19	0.23	0.23
author/works_written	27	0.54	0.54	0.56	0.47	0.65	0.68	0.62	**0.76**
parent/child	20	0.13	0.24	0.58	0.44	0.72	0.74	0.79	**0.80**
person/place_of_birth	20	0.70	0.67	0.74	0.45	0.75	0.73	0.73	**0.77**
person/place_of_death	19	0.79	0.79	0.86	**0.89**	0.83	0.85	0.86	0.85
neighborhood/neighborhood_of	11	0.00	0.00	0.09	0.47	0.70	**0.72**	0.67	0.69
person/parents	6	0.28	0.32	0.67	0.64	0.61	**0.68**	0.55	0.64
company/founders	4	0.25	0.25	0.53	0.24	0.77	**0.80**	0.64	0.69
sports_team/league	4	0.00	0.43	0.18	0.21	0.59	**0.70**	0.56	0.48
film/directed_by	3	0.08	0.19	0.33	0.12	0.34	**0.35**	0.12	0.11
team_owner/teams_owned	2	0.00	0.50	*0.70*	0.55	0.38	0.61	*0.70*	0.61
team/arena_stadium	2	0.00	0.08	0.08	0.04	0.13	0.13	0.12	**0.15**
roadcast/area_served	2	*1.00*	0.50	*1.00*	0.58	0.58	0.83	0.83	*1.00*
structure/architect	2	0.00	0.00	*1.00*	0.27	*1.00*	*1.00*	*1.00*	*1.00*
composer/compositions	2	0.00	0.00	0.00	0.50	0.67	**0.83**	0.50	0.57
person/religion	1	0.00	*1.00*	*1.00*	0.50	*1.00*	*1.00*	*1.00*	*1.00*
film/produced_by	1	*1.00*	*1.00*	*1.00*	*1.00*	0.50	0.50	*1.00*	0.33
MAP		0.32	0.42	0.56	0.46	0.62	**0.67**	0.65	0.64
Weighted MAP		0.48	0.50	0.57	0.52	0.67	0.68	0.68	**0.70**

To verify what benefit can be obtained with our low-rank assumption on the relation interaction coefficients, and to investigate whether this may have somehow hurt the model's general performance, we ran our system on the full training portion of the dataset. Table 17 presents the results for several models from the literature (M09: Mintz et al. (2009b), Y11: Yao et al. (2011), S12: Surdeanu et al. (2012), R13-*: Riedel et al. (2013a)), our FM-based implementation (FM) and our FM implementation using contextual neighbourhood information (FM+n). It shows that we can indeed obtain gains in terms of wMAP when contextual neighbourhood information is incorporated with a low-rank assumption on the neighbourhood weights of relations. The MAP score of FM+n is lower but note that this is mostly due to relations with very few true facts in the test pool that outweigh the benefits on the rest. Note that the results we report here are not comparable with those reported by Petroni et al. (2015) as they used a different version of the dataset with different pre-processing and additional metadata.

Table 18 presents the relations with the most surface patterns in the training corpus. Observe that the relations that model $FM + n$ improved on FM the most in Table 17 tend to rank high on this table, e.g., *author/works_written* or *person/place_of_birth*, thus demonstrating that surface patterns are useful for better modelling of the knowledge base relations.

We now turn our attention to the one-shot learning experimental setup. Figure 10 presents the results of one-shot experiments for the two variants of our model (FM and FM+n) and R13-F from Riedel et al. (2013b). The figure shows that the difference in performance between models FM and FM+n is wider when less supervision data is available. These results demonstrate that the

Table 18: Number of associated surface pattern fact mentions of each relation in the training set

Relation	# patterns
location/location/containedby	786
business/person/company	332
people/person/nationality	235
book/author/works_written	229
people/person/place_of_birth	216
people/place_of_death	117
organization/parent/child	77
location/neighborhood_of	74
film/film/directed_by	25
business/company/founders	25
sports/sports_team/arena_stadium	23
sports/teams_owned	19
people/person/religion	16
film/film/produced_by	15
people/person/parents	12
sports/sports_team/league	9
broadcast/broadcast/area_served	7
architecture/structure/architect	7
music/composer/compositions	6

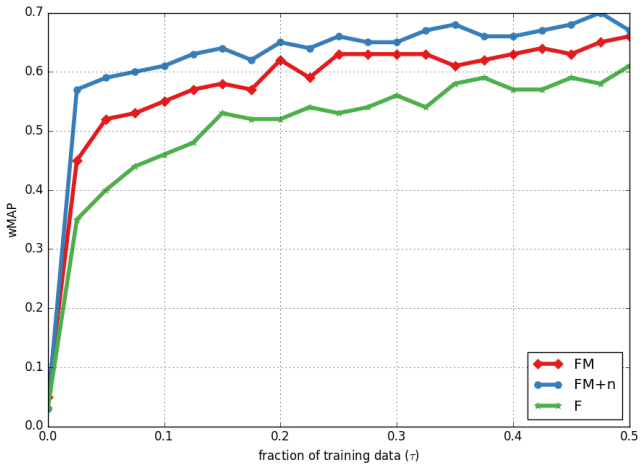

Figure 10: One-shot comparison between FM, FM with contextual neighbourhood features (FM+n) and Model R13-F.

contextual neighbourhood information incorporated by model FM+n enhanced its performance when less supervision labels are available to the model.

Figure 11 presents results of our best model (FM+n) compared to state-of-the-art models from Rocktäschel et al. (2015b) (R15-Joint) and Demeester et al. (2016) (D16-FSL). Note that our system does not make use of any rules as extra supervision data, and this affected its performance in the zero-shot setting. Nevertheless, it was still able to obtain better coverage, as measured by the

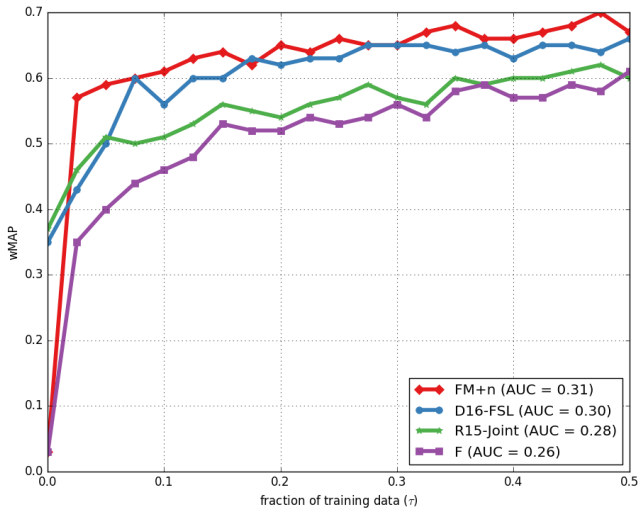

Figure 11: One-shot comparison against previous work. Results obtained from Demeester et al. (2016).

wMAP AUC.

5.6 Related work

Related to our work are various approaches to relation extraction and knowledge base population with universal schemas. For instance, Toutanova et al. (2015) and Verga et al. (2016) extended a universal schema-approach to model the compositional nature of textual representations. Instead of learning a single representation for each surface pattern, Toutanova et al. (2015) and Verga et al. (2016) learn a representation which is a composition of the lexical items which make up the surface patterns. While Toutanova et al. (2015) made use of a convolutional neural network as compositional operator, Verga et al. (2016) explored several other operators, including the use of recurrent neural networks and bidirectional Long Short-Term Memory networks (LSTMs). This enables textual representations with lexical overlap to share statistical strength, and therefore enhance the generalization ability of the model to unseen textual patterns. Various combinations of matrix and tensor factorization models for universal schema-based relation extraction were explored in Singh et al. (2015).

5.7 Conclusion

We considered the task of learning to extract relations with few annotated labels. We proposed a FM based model that utilized contextual surface patterns, that is readily available within the text itself. We showed that our approach improved in extraction accuracy compared to previous approaches. While we have represented the contextual neighbourhood information by a single low-rank representation for each surface pattern, a future direction for our work is investigating the use of compositional representations for the surface patterns, which have been shown to lead to better modelling of knowledge base relations (Toutanova et al., 2015; Verga et al., 2016).

5.8 Publications

The work reported in this chapter is currently under submission.

6 Jack The Reader

Knowledge Base Construction is a broad topic, containing a range of diverse tasks and methods. Ideally, we would like the SUMMA platform to give users the appropriate flexibility in choosing the appropriate approach for their purposes. Jack The Reader (JTR) is a new open source framework for Machine Reading, covering a range of tasks including Knowledge Base Population, Question Answering and Textual Entailment. Our intention is to deliver the KBP components of SUMMA within the Open Source JTR framework, giving SUMMA users access to a growing library of models.

In the remainder of this section we will describe the JTR framework and describe the possible benefits of cross-fertilisation across these diverse tasks.

6.1 Motivation

Machine reading is an increasingly important area of NLP research that focuses on high-level tasks which require deep natural language understanding. More established domains with large numbers of researchers often employ standard toolkits that allow re-use and re-combination of components to speed research and development. For example, Moses (Koehn et al., 2007) and Nematus (Sennrich et al., 2017) play this role for Machine translation research while Kaldi (Povey et al., 2011) is commonly used in Speech Recognition. Currently however a lack of shared frameworks for Machine Reading tasks means that much research effort is absorbed by the repeated redevelopment of equivalent pipeline components, such as pre- and post-processing routines or sentence encoders/-decoders. The ability to re-use and re-combine these and other structures would not only eliminate some of the drudgery of model development, but would also promote research progress by facilitating the evaluation of existing models on new datasets and the innovation of new models by combining existing modules. Moreover, in the context of SUMMA, adopting a single framework for all KBP models should simplify the task of integrating with the rest of the platform.

For these reasons, we propose to employ the JTR framework for the SUMMA KBP deliverables. NLP model architectures are specified in Python and TensorFlow, while a standard JSON format facilitates dataset importing and preprocessing. Moreover, its modular architecture allows the definition of a new machine reading pipeline from scratch with only a few lines of Python code, and enables users to quickly evaluate new or existing models on a variety of datasets with very little effort. Already, JTR covers a variety of different machine reading tasks, including (Multiple-Choice, Extractive and Generative) Question Answering, Knowledge Base Population (KBP) and Recognising Textual Entailment (RTE).

6.2 Overview of JTR

The goal of JTR is to facilitate re-use of Machine Reading components and encourage evaluation across multiple datasets. To achieve this, we have attempted to modularise key elements of typical pipelines. Definition of a standard data format, JF, means that once a dataset has been converted, it is ready to be used by any of the JTR models relevant to that task. Within JTR itself, a modular structure supports a building block approach to model construction. In particular, data flow between these modules is specified in terms of TensorPorts which define the abstract structure of expected inputs and outputs, serving to maintain a consistent architecture.

Table 19: Models currently available in JTR.

Task	Model	
KBP	Model F	Riedel et al. (2013a)
	TransE	Bordes et al. (2013)
	DistMult	Yang et al. (2015)
	Complex	Trouillon et al. (2016)
QA	FastQA	Weissenborn et al. (2017)
	BOW	Weissenborn et al. (2017)
RTE	Conditional Bi-LSTM	Rocktäschel et al. (2015a)
	DAM	Parikh et al. (2016)
	ESIM	Chen et al. (2016)

6.3 JTR architecture

The core of the JTR framework is the JTReader Class which integrates input, model and output modules, co-ordinating batching, training and evaluation processes. To encourage re-usability, a unifying data format (JF) has been designed to encompass a wide range of Machine Reading tasks within a single scheme. In addition to these general Machine Reading structures, a number of specific utilities and modules are included, which can be composed to implement various state-of-the-art systems for a range of tasks.

This composition is facilitated by JTR's system of TensorPorts which act as connectors between the modules. A given TensorPort defines the abstract structure of data flowing between modules - e.g. whether answers are single boolean values or lists of ids representing tokens - and this serves to ensure that modules are joined in a consistent architecture.

6.3.1 JF format

The JTR data format (JF) is a JSON-based format which defines the contents of single data instances. An instance is characterised by:

- One or more *questions* (SINGLE, MULTIPLE).
- One or more *answers* (SINGLE, MULTIPLE).
- Optional supporting evidence (*support*) from which the answer can be extracted or inferred (NONE, SINGLE, MULTIPLE).
- Optional per-instance or global answer *candidates* (NONE, PER-CANDIDATE, GLOBAL).

Typically, an *answer* is the target to be predicted, while the *question* and *support* are text inputs from which the prediction is to be made. However, the framework's flexibility allows a great deal of freedom in defining and using these structures. KBP tasks, for example, can be encoded in a variety of ways. One approach is to treat entity pairs as the *answers* to *questions* defined by KB relations and surface patterns. Alternately, *questions* can be whole triples of an entity pair linked by a relation, to which the *answer* is a binary True or False value. In either case, *support* can

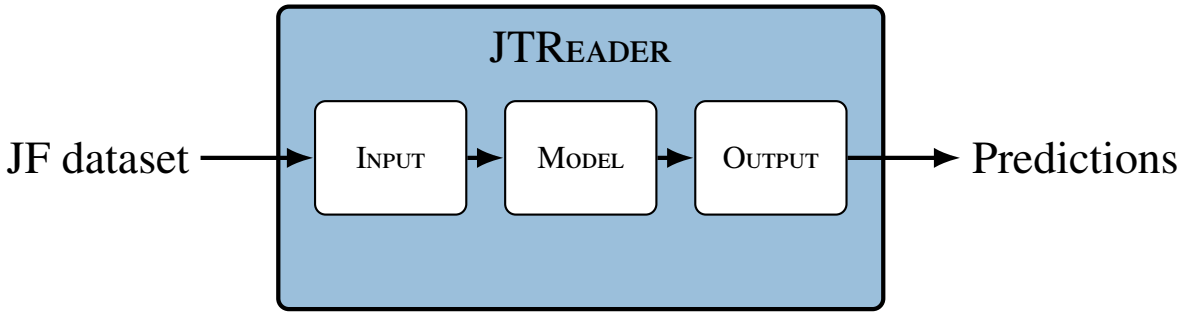

Figure 12: JTR overview: Flow of data through JTR

carry additional information (e.g. concerning the textual context of entity mentions or the network structure relevant to a particular relation) that aids prediction.

6.3.2 Input, model and output Modules

- Input Module: responsible for pre-processing the data (e.g. feature construction, tokenisation and mapping tokens to ids) and defining the batch structure.
- Model Module: defines the output function and loss in Tensorflow.
- Output Module: handles the processing required to turn model outputs back into the appropriate format for answers (e.g. mapping lists of tokenids back into sentences).

6.3.3 JTReader

In Figure 12 we give an overview of how the data format and the core components interact with each other.

A JTReader is a wrapper around given input, model and output modules and as such defines a specific machine reading system. It controls the flow of data between the modules, defining a training loop which optimises the loss defined in the model module over batches supplied by the input module. During training progress is monitored by a set of performance and evaluation hooks, which run at specified times (e.g. every 100 iterations, every epoch or after training) and on completion the output module can generate predictions for a test dataset.

6.4 JTR in SUMMA

Delivering KBP functionality in JTR has a number of benefits. Adopting a single architecture will simplify the process of switching between models, when that is needed, because the interfaces to the SUMMA platform will be isolated in separate input and output modules. Moreover, the modular and open-source nature of the framework should provide a growing library of model modules that users can easily swap in and out of use. Beyond usability, the modular structure should also aid research by allowing re-use of existing components. It should therefore speed the creation of new models and their evaluation across multiple datasets.

We also foresee that working within a general Machine Reading framework is likely to be beneficial. Elements of the RTE and QA tasks are likely to become increasingly important to KBP.

Intelligent relation extraction should be guided by the same logical rules and constraints that apply to the RTE task. For example, Demeester et al. (2016) demonstrate that KBP performance can be enhanced by injecting common-sense implications into the structure of relation embeddings. Equally, the answer to a query should be the same whether it is applied to source documents or a KB derived from them. In fact, QA systems are commonly applied to the slot-filling task of TAC-KBP (Ji and Grishman, 2011).

7 Conclusion

This deliverable described the initial progress in automatic knowledge base construction and the related technologies of entity tagging and linking. Our immediate plans is to participate in the upcoming Text Analytics Conference-Knowledge Base Population challenge using the technologies developed and keep developing them further in the second half of the project. In the context of this effort we are currently integrating the factorization machine approach to relation extraction with the entity recognition and linking componenets developed in the context of a research visit of Abiola Obamuyide (USFD) at the Priberam. Furthermore, UCL is currently working on developing a TAC-KBP module using the Jack The Reader software infrastructure.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 85–94, New York, NY, USA, 2000. ACM. ISBN 1-58113-231-X. doi: 10.1145/336597.336644. URL http://doi.acm.org/10.1145/336597.336644.

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

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics, 2015. ISBN 978-1-941643-72-3. URL http://aclweb.org/anthology/P/P15/P15-1034.pdf.

Anders Björkelund and Jonas Kuhn. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 47–57, 2014. URL http://aclweb.org/anthology/P/P14/P14-1005.pdf.

Phil Blunsom and Trevor Cohn. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72. Association for Computational Linguistics, 2006.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008a. ISSN 07308078.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008b. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376746. URL http://doi.acm.org/10.1145/1376616.1376746.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In Christopher J. C. Burges et al., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2787–2795, 2013.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Machine Learning*, 94(2):233–259, 2014.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, AAAI Spring Symposium Series*, 2015.

Razvan C Bunescu and Raymond J Mooney. Subsequence kernels for relation extraction. *Advances in Neural Information Processing Systems*, 18:171, 2006. ISSN 1049-5258.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038, 2016. URL http://arxiv.org/abs/1609.06038.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 551—-561, Austin, TX, USA, 2016. Association for Computational Linguistics.

Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011a. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078186.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011b.

Savelie Cornegruta and Andreas Vlachos. Timeline extraction using distant supervision and joint inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1936–1942, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1200.

Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press, 1999. ISBN 1-57735-083-9. URL http://dl.acm.org/citation.cfm?id=645634.663209.

Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning,* Prague, Czech Republic, 28–30 June 2007, pages 708–716, 2007.

Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, 4(Table 2): 423–es, 2004. doi: 10.3115/1218955.1219009. URL http://portal.acm.org/citation.cfm?doid=1218955.1219009.

Hal Daumé III and Daniel Marcu. Learning as search optimization: approximate large margin methods for structured prediction. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, pages 169–176, 2005. doi: 10.1145/1102351.1102373. URL http://doi.acm.org/10.1145/1102351.1102373.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Daniel D. Lee et al., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation embeddings. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1389–1399. The Association for Computational Linguistics, 2016. ISBN 978-1-945626-25-8. URL http://aclweb.org/anthology/D/D16/D16-1146.pdf.

Leon Derczynski. *Determining the Types of Temporal Relations in Discourse*. PhD thesis, University of Sheffield, 2013.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. arXiv e-print arXiv:1707.01476 [cs.LG], University College London, July 2017. URL https://arxiv.org/abs/1707.01476.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In Sofus A. Macskassy et al., editors, *The 20th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pages 601–610. ACM, 2014. ISBN 978-1-4503-2956-9.

David K. Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In Corinna Cortes et al., editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2224–2232, 2015.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. ISSN 01628828.

Eraldo R. Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *EMNLP-CoNLL Shared Task*, pages 41–48. ACL, 2012.

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

David Ferrucci and Adam Lally. Uima: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, September 2004. ISSN 1351-3249. doi: 10.1017/S1351324904003523. URL http://dx.doi.org/10.1017/S1351324904003523.

Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*, 2017.

Guillaume Genthial. Sequence Tagging and Named Entity Recognition with Tensorflow (LSTM + CRF), 2017. URL https://github.com/guillaumegenthial/sequence{_}tagging.

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

Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 765–774, 2011. doi: 10.1145/2009916.2010019. URL http://doi.acm.org/10.1145/2009916.2010019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 30–34, 2013. URL http://aclweb.org/anthology/P/P13/P13-2006.pdf.

Benjamin Heinzerling, Alex Judea, and Michael Strube. Hits at tac kbp 2015: Entity discovering and linking, and event nugget detection. In *Proceedings of the Eighth Text Analysis Conference (2015). Gaithersburg, MD: National Institute of Standards and Technology*, page 109, 2015.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 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, 2011.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167*, 2015.

Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1148–1158, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002618.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. Overview of the tac 2011 knowledge base population track. In *Proceedings of Text Analysis Conference (TAC)*, 2011.

Heng Ji, Joel Nothman, Hoa Trang Dang, and Sydney Informatics Hub. Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*, 2016.

Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 655–665. The Association for Computer Linguistics, 2014. ISBN 978-1-937284-72-5.

Yoon Kim. Convolutional Neural Networks for Sentence Classification. In Alessandro Moschitti et al., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014. ISBN 978-1-937284-96-1.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*, 2015.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Denis Krompaß, Stephan Baier, and Volker Tresp. Type-Constrained Representation Learning in Knowledge Graphs. In Marcelo Arenas et al., editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 640–655. Springer, 2015. ISBN 978-3-319-25006-9.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 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, 2009.

Egoitz Laparra, Itziar Aldabe, and German Rigau. Document level time-anchoring for timeline extraction. In *ACL*, 2015.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence ({AAAI-08})*, pages 646–651, 2008. ISBN 978-1-57735-368-3.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Comput. Linguist.*, 39(4):885–916, December 2013. ISSN 0891-2017. doi: 10.1162/COLI_a_00152. URL http://dx.doi.org/10.1162/COLI_a_00152.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1135.

Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219745. URL http://doi.acm.org/10.1145/219717.219745.

Hector Llorens, Estela Saquete, and Borja Navarro. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291. Association for Computational Linguistics, 2010.

Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*, 2016. ISSN 1098-6596. doi: 10.18653/v1/P16-1101. URL http://arxiv.org/abs/1603.01354.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org, 2015.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL: System Demonstrations*, pages 55–60, 2014.

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

André FT Martins and Mariana SC Almeida. Priberam: A turbo semantic parser with second order features. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 471–476, 2014.

David McClosky and Christopher D Manning. Learning constraints for consistent timeline extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 873–882. Association for Computational Linguistics, 2012.

E.G. Miller, N.E. Matsakis, and P.a. Viola. Learning from one example through shared densities on transforms. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, 1(1), 2000. ISSN 1063-6919.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2132.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA, 2009a. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL http://dl.acm.org/citation.cfm?id=1690219.1690287.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA, 2009b. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL http://dl.acm.org/citation.cfm?id=1690219.1690287.

Patrick AP Moran. Notes on Continuous Stochastic Phenomena. *Biometrika*, 37(1/2):17–23, 1950.

Bilel Moulahi, Jannik Strötgen, Michael Gertz, and Lynda Tamine. Heideltoul: A baseline approach for cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 825–829, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2139.

Borja Navarro and Estela Saquete. Gplsiua: Combining temporal information and topic modeling for cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 820–824, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-2138.

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1414–1420. AAAI Press, 2007. ISBN 978-1-57735-323-2. URL http://dl.acm.org/citation.cfm?id=1619797.1619872.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In Lise Getoor et al., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress, 2011.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016a.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic Embeddings of Knowledge Graphs. In Dale Schuurmans et al., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1955–1961. AAAI Press, 2016b. ISBN 978-1-57735-760-5.

Mathias Niepert. Discriminative Gaifman Models. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3405–3413, 2016.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

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. Summa at tac knowledge base population task 2016. In *Proceedings of the Text Analysis Conference -TAC*, pages 1–9, Gaithersburg, Maryland USA, 2017.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 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*, 2015.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016. URL http://arxiv.org/abs/1606.01933.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation, 2014. URL https://nlp.stanford.edu/projects/glove/.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. CORE: Context-Aware Open Relation Extraction with Factorization Machines. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1763–1773, 2015. ISBN 9781941643327.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea, 2012.

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

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 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, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002642.

Steffen Rendle. Factorization machines. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 995–1000, 2010. ISSN 15504786.

Steffen Rendle. 57 Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol. Article*, 3(22), 2012. doi: 10.1145/2168752.2168771.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009. ISSN 1469493X.

Sergio J Rey and Luc Anselin. PySAL: A Python Library of Spatial Analytical Methods. *Handbook of applied spatial analysis*, pages 175–193, 2010.

Sebastian Riedel, Limin Yao, Andrew Mccallum, and Benjamin M Marlin. Relation Extraction with Matrix Factorization and Universal Schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2013a.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation Extraction with Matrix Factorization and Universal Schemas. In *Proceedings of NAACL/HLT*, pages 74–84, 2013b.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664, 2015a. URL http://arxiv.org/abs/1509.06664.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. *North American Association for Computational Linguistics*, pages 1119–1129, 2015b.

Evan Sandhaus. The New York Times Annotated Corpus, 2008.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. *arXiv preprint arXiv:1703.06103*, 2017.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch-Mayne, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Miceli Barone, Jozef Mokry, and Maria Nadejde. *Nematus: a Toolkit for Neural Machine Translation*, pages 65–68. Association for Computational Linguistics (ACL), 4 2017. ISBN 978-1-945626-34-0.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In Chin-Wan Chung et al., editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, pages 373–374. ACM, 2014. ISBN 978-1-4503-2745-9.

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

Sameer Singh, T Rocktäschel, and Sebastian Riedel. Towards Combined Matrix and Tensor Factorization for Universal Schema Relation Extraction. *Naacl*, pages 135–142, 2015. URL http://rockt.github.io/pdf/singh2015towards.pdf.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Jannik Strötgen, Julian Zell, and Michael Gertz. Heideltime: Tuning english and developing spanish resources for tempeval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 15–19, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S13-2003.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

Mihai Surdeanu and Heng Ji. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proceedings of the 2014 Text Analytics Conference*, 2014.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance Multi-label Learning for Relation Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP '12*, pages 455–465, 2012. ISBN 9781937284435.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

Kristina Toutanova and Danqi Chen. Observed Versus Latent Features for Knowledge Base and Text Inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing Text for Joint Embedding of Text and Knowledge Bases. In *EMNLP*, volume 15, pages 1499–1509, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016. URL http://jmlr.org/proceedings/papers/v48/trouillon16.html.

Naushad UzZaman and James F. Allen. Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 351–356, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6. URL http://dl.acm.org/citation.cfm?id=2002736.2002809.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S13-2001.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew Mccallum. Multilingual Relation Extraction using Compositional Universal Schema. *arXiv*, pages 1–15, 2016.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics, 2007.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics, 2010.

Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269, April 1967.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Fastqa: A simple and efficient neural architecture for question answering. *CoRR*, abs/1703.04816, 2017. URL http://arxiv.org/abs/1703.04816.

Jun Xie, Chao Ma, Janardhan Rao Doppa, Prashanth Mannem, Xiaoli Z. Fern, Thomas G. Dietterich, and Prasad Tadepalli. Learning greedy policies for the easy-first framework. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2339–2345, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9594.

I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. Learning Distributed Representations of Texts and Entities from Knowledge Base. *ArXiv e-prints*, May 2017.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations (ICLR)*, 2015.

Limin Yao, Sebastian Riedel, and Andrew Mccallum. Structured Relation Discovery using Generative Models. *Technology*, pages 1456–1466, 2011. URL http://aclweb.org/anthology-new/D/D11/D11-1135.pdf.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. Learning Discriminative Projections for Text Similarity Measures. In Sharon Goldwater et al., editors, *Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*, pages 247–256. ACL, 2011. ISBN 978-1-932432-92-3.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003. ISSN 1532-4435.

ENDPAGE

SUMMA

H2020-ICT-2015 688139

D4.1 Initial Progress Report on Automatic Knowledge Base Creation