Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4501–4508
Marseille, 11–16 May 2020

Hedwig: A Named Entity Linker

Marcus Klang, Pierre Nugues
Lund University
Department of Computer Science
{marcus.klang, pierre.nugues}@cs.lth.se

Abstract

Named entity linking is the task of identifying mentions of named things in text, such as "Barack Obama" or "New York", and linking these mentions to unique identifiers. In this paper, we describe Hedwig, an end-to-end named entity linker, which uses a combination of word and character BILSTM models for mention detection, a Wikidata and Wikipedia-derived knowledge base with global information aggregated over nine language editions, and a PageRank algorithm for entity linking. We evaluated Hedwig on the TAC2017 dataset, consisting of news texts and discussion forums, and we obtained a final score of 59.9% on CEAFmC+, an improvement over our previous generation linker Ugglan, and a trilingual entity link score of 71.9%.
Keywords: named entity recognition, named entity linking, named entity annotation

1. Introduction

Named entity linking (NEL) is the task of automatically finding and linking mentions of things to unique identifiers. The word *thing* is too broad for the linkage problem; a more concrete definition used in this paper is linking uniquely separable things, which we can identify by a *name*, i.e. named entities. The classes of *named entities* we will then try to link are instances of persons, organizations, locations, etc.

Take for instance the named entity of class location: "New York". This mention can refer to the state[1] of New York or the large city[2] situated in that particular state. For the latter, a matching unique identifier could be the English Wikipedia label: `New_York_City`.

A typical NEL pipeline consists of many phases including a name finding, mention detection (MD) phase (e.g. detecting "New York" in a text), a candidate generation (CD) phase (e.g. state or city), and an entity linking (EL) phase (e.g. assigning the label). In addition, these phases might be defined independently (Cucerzan, 2007), or trained jointly (Ganea and Hofmann, 2017). The MD phase is frequently a named entity recognizer (NER), which finds and classifies spans of strings in a set of predefined classes such as persons, organization, location, etc. The CD phase uses the classified mention as input, possibly with context, and from this information generates a list of entity candidates. Finally, the entity linking phase ranks and selects the most probable or coherent set of candidate entities. It assigns each mention a label, which corresponds to the unique identifier of the the selected candidate.

The unique identifier can be local or global, and its concrete format is determined by the linker method, which can span the spectrum of fully supervised to unsupervised. This paper uses a supervised approach by linking to predetermined identifiers. These identifiers are provided by an entity repository which we refer to as the *knowledge base* (KB).

EDL annotated datasets use different kinds of corpora and come with different evaluation procedures. This paper will present a named entity linker for the 2017 edition of the Text Analysis Conference (TAC) Entity Discovery and Linking (EDL) task with its provided benchmark (Ji et al., 2017). This task was selected because it provides a multilingual gold standard. This dataset is diverse in its content and is a combination of real-world noisy texts found on the internet. This type of dataset presents challenges that all entity linkers would encounter when applied in the real world.

1.1. TAC EDL 2017

The TAC EDL task consists of linking two categories of mentions:

- Named mentions divided into five classes:
  - `PER`, Persons (nonfictional)
  - `ORG`, Organizations, (companies, institutions, etc.)
  - `LOC`, Location (natural locations, such as mountains, oceans, lakes etc.)
  - `GPE`, Geopolitical entities (cities, administrative areas, countries, states, municipalities, etc.)
  - `FAC`, Facilities (airports, transportation infrastructure, man-made buildings, hospitals etc.)
- Nominal mentions, involving a limited set of common nouns coreferring with a named mention. For instance, *Barack Obama* would be the named mention and the nominal mention would be *president*. Other common relations are *son*, *wife*, *daughter*, *father*, *company*, *area*, etc. The nominal mentions are classified and linked in the same manner as named mentions.

The corpus is a mixture of discussion forum (DF) and newswire (NW) text in three languages: English, Spanish, and Mainland Chinese stored in XML and HTML (2014). The gold standard provides links to the Freebase KB and out-of-KB labels. These latter labels start with "NIL" followed by a number which spans all three languages. The Freebase identifiers are connected in the BaseKB provided

[1] `https://en.wikipedia.org/wiki/New_York_(state)`
[2] `https://en.wikipedia.org/wiki/New_York_City`

by TAC. The final score is based on the performance on all three languages.
We subdivided the corpus into a training and test set based on years. The 2017 dataset is the test set, and 2014-2016 is our training set. It is important to note that the 2014 edition does not have Spanish or Chinese texts.
Nominal mentions were added in 2015, but they were not fully annotated until 2016 which means scarce data is available for training nominal detection using deep models.

1.2. Specifics and Limitations to Hedwig

Hedwig uses data and statistics from Wikipedia, almost exclusively, and as such, it is natural for us to use Wikidata as the primary KB. Wikidata provides unique identifiers in the form of Q-numbers, e.g. "Barack Obama", the president, has identifier: `Q76`. Wikidata binds the Wikipedia languages editions together and is continuously updated. It is thus more up-to-date than other repositories, making it the most logical choice for us.
To be compliant with the TAC EDL gold standard, we converted the Q-numbers into Freebase using a mapping[3] provided by Google, produced at the archiving and termination of Freebase as a public open knowledge base. This mapping is not perfect: A few Q-numbers are represented by multiple Freebase entries. We resolved them heuristically using the lowest Q-number.

2. Related Work

Multilingual named entity recognition has its modern roots with the CoNLL03 task of Language-Independent Named Entity Recognition (Tjong Kim Sang and De Meulder, 2003), where the best model used simple linear classifiers (Florian et al., 2003). Neural models, starting with feed forward architectures, improved the recognition performance. Examples of such models include Collobert et al. (2011) and the exponential weight encoding method (Fixed-Size Ordinally-Forgetting Encoding, FOFE) (Zhang et al., 2015). These architectures were ultimately surpassed by deeper recurrent neural models using LSTMs (Hochreiter and Schmidhuber, 1997) and CRFs in different combinations with or without word character encoders (Chiu and Nichols, 2016; Ma and Hovy, 2016). Straková et al. (2019) proposed a sequence to sequence model with attention, converting an input token sequence into a label sequence. This sequence-to-sequence model makes use of multiple contextualized word embeddings as input. Finally, a recent system by Jiang et al. (2019) improved the state-of-the-art performance on the CoNLL03 task with a differential neural network search method.
Word embeddings are a key ingredient to NER; the most commonly used word embedding started with Mikolov et al. (2013), followed by Pennington et al. (2014), to more recent developments by Mikolov et al. (2018), and deeper models by Peters et al. (2018). Flair (Akbik et al., 2019) is a recent NLP framework that provides a simplified interface to many state-of-the-art word embeddings.
Modern entity linking uses a variety of methods such as simple classification models (Bunescu and Paşca, 2006; Cucerzan, 2007; Milne and Witten, 2008), end-to-end linkage with a voting scheme for linkage (Ferragina and Scaiella, 2010), graphical models (Hoffart et al., 2011; Guo and Barbosa, 2014), integer linear programming (Cheng and Roth, 2013), fully probabilistic models (Ganea et al., 2016) to deeper neural models (Ganea and Hofmann, 2017). Wainwright et al. (2008)[4] proved that entity linkage which tries to maximize local and global agreement jointly during linkage is NP-hard to solve, which most authors approximate or simplify to reach feasibility.
The system described in this paper, Hedwig, builds on Klang and Nugues (2018) with significant improvements and contributions in four areas:

1. Updated data sources and preprocessing;
2. A named entity recognizer based on a BILSTM neural network architecture;
3. The linker considers a larger candidate graph with more features, and an improved PageRank solver;
4. Finally, the introduction of an entity type mapping that can remove unwanted entities with no relevance to the evaluation.

3. Data

In the making of Hedwig, we used these data sources:

- Nine Wikipedia editions: en, es, fr, de, sv, ru, zh, da, no, scraped using the Wikipedia REST API[5] in October 2018.
- Wikidata JSON dump from October 2018
- TAC EDL Data 2014-2016
- Manually annotated mappings of classes in Wikipedia to a set of predefined classes.

3.1. Wikidata: Our KB

The Wikidata JSON dump is delivered as one large gzip or bzip2. This file when decompressed is one single JSON object; it does however use "one JSON object per line" approach for easier processing. Using standard bash tools, we split it into multiple parts with 50,000 objects per file to enable efficient cluster processing. We converted this dump with a Wikidata parser, which transformed the JSON dump into Parquet files for further processing and information extraction. The information we converted was: Q-number, description, alias, claims also known as properties and sitelinks. A subset of most common claim datatypes are supported; the rest is either ignored or encoded as plain strings.

3.2. Wikipedia

We scraped the nine editions using the REST API by first downloading a list of page names from Wikimedia's dump

[3] `https://developers.google.com/freebase/`

[4] cited in Globerson et al. (2016).

[5] `https://[lang].wikipedia.org/api/rest_v1/`

site[6], more specifically the [lang]wiki-[date]-all-titles-in-ns0.gz file which contains all page labels[7]. We then processed this file in sequence, where we downloaded pages in parallel with a custom Python tool. This list of page names is slightly out of sync with the online version, resulting in some nonexisting pages; pages not found were ignored. Pages that failed to download due to server errors or network failure were retried once more at the end of the scraping process. In total, the retries amounted to 5-250 pages; most of these pages were eventually retrieved.
The REST API provided by Wikimedia is using Parsoid[8] internally and produces HTML outputs with added metadata tags and attributes for the elements. This parser does not produce an exactly identical structure to the publicly facing rendering. However, when sampling pages, no visible differences to the content or format was significantly evident. For the Chinese version, a translation table was included in the output to convert the character sequences between the different variants: Hong Kong, Singapore, etc.

4. Data Preprocessing

4.1. Wikipedia

Wikipedia is refined in two steps:

1. Import, converting HTML to Docria layers;
2. Link, resolving Wikipedia anchors to Wikidata.

In the import step, we parse HTML using JSoup[9], which converts the raw HTML string into a structured Document Object Model (DOM). We used rules applied recursively to the DOM tree to filter out the markup and produced a flattened document consisting only of readable text. In addition, during the flattening process or tree traversal, enough information was retained to produce multiple layers of spans with added metadata covering paragraphs, sections, anchors, lists, tables, italic, bold etc. These layers were stored using Docria (Klang and Nugues, 2019) that can represent this type of data and store exact string mappings. With the exception of the Chinese version, all the editions were parsed identically. The Chinese version required a pre-conversion step to produce a Mainland Chinese version using the provided translation table; this conversion might not be perfectly accurate.
The link step converts all the anchors in the processed Wikipedia dump into Q-numbers using sitelinks in the Wikidata dump. It is necessary to do this link step for two reasons: links in Wikipedia refer to pages using titles not unique identifiers, and many links are placeholder suggestions for future articles.
Ultimately, this produces a Docria dump which is fully linked to Wikidata, and contains enough semantic information and structure for all further processing.
We chose nine editions to increase the statistical basis for entity-to-entity co-occurrences. Links in the Docria dump consist of multilingual Wikidata identifiers. Because of this, they can provide prior information that may benefit lower resourced languages. Information from all nine languages are only used in the linking stage.

4.2. Word Embeddings

We trained the word embeddings on a 2016 version of the English, Spanish, and Chinese Wikipedia using the `word2vec` tool released by Mikolov et al. (2013).

4.3. Entity Classification

Many articles in Wikipedia do not conform to acceptable classes in TAC EDL. To improve entity linkage precision and reduce noise, we pruned the entity candidates based on their class. To determine the class of an entity, we used a classifier that we trained on a dataset extracted from Wikidata.
Using the TAC Annotation guidelines as a basis, we wrote rules to sample a diverse set of entities with usable instance-of relations. We mapped them to nine classes:

PER, natural persons, humans

PER_F, fictional characters or persons

LOC, natural locations, continents, lakes, rivers, mountains, streams, etc.

GPE, geopolitical entities, countries, administrative areas, regions, etc.

ORG, organizations, institutions, business entities etc.

EVT, events, sport events, conflicts, wars, etc.

WRK, products, newspaper, books, films, TV series, etc.

FAC, man made buildings, transportation infrastructure, airports, hospitals, landmarks, etc.

NONE, all that does not fall into any above. This explicitly includes: wikipedia categories, templates, disambiguation pages, etc.

The input features for the model were:

- Direct relations such as P31 (instance-of) Q5 (human)
- Boolean indicating the existence of a P31 instance. Empirically, most entities without an instance-of are exceptionally hard to disambiguate and should, most of the time, be classified as `NONE`.
- All path segments from the initial seed entity with source, relation, and target, to the depth of 5.

We used a *breadth first search* (BFS) to the depth of 5 to find suitable segments by following edges for relations: 31 (instance-of), 279 (subclass-of), 1269 (facet-of) and 1889 (different-from). The training data consists of seed entities mapped to one of these nine classes, and all features were generated on the fly. To produce negative examples, all unclassified are assumed to have label `NONE`. To balance positive and negative bias, the negative share was set to 10% per mini-batch. In practice, this makes the classifier default to `NONE` for all unknown relations; the large

[6] `http://dumps.wikimedia.org/`
[7] `Wikipedia name for article titles`
[8] `https://www.mediawiki.org/wiki/Parsoid`
[9] `https://jsoup.org/`

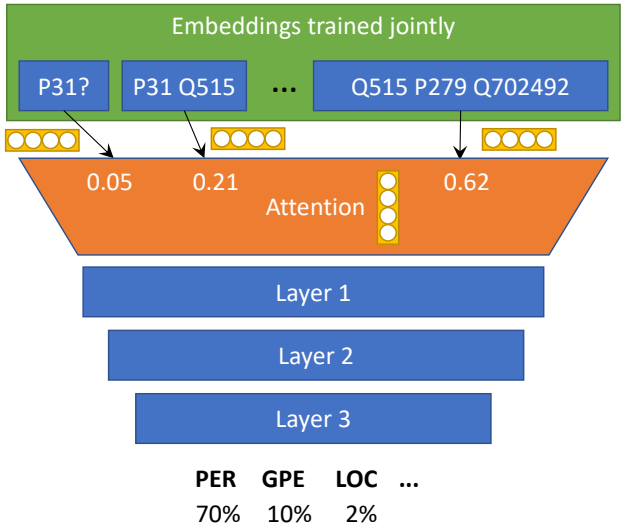

Figure 1: Entity classifier neural network, P31 is instance-of, Q515 is "city" entity and Q702492 is "urban area" which is a subclass of (P279) city.

positive share should force the classifier to learn given examples, and retain the most significant patterns common to each class. Phrased differently, we favored precision over recall in this setting.

We trained a three layer neural model with a simplified attention mechanism to classify types as shown in Figure 1. This model treats all input features as a trainable embedding, a simplified attention mechanism is used to produce a single vector of all inputs through a weighted average produced by the attention mechanism and then three RELU layers with a final softmax layer to predict the output.

5. Mention-Entity Dictionary

5.1. Mention Dictionary

We created a mention-to-entity dictionary from all anchor texts, aliases, and titles of Wikipedia articles, where for each mention, we extracted the entity candidates.

This dictionary is associated with a tokenizer based on a JFlex parser and described in Sect. 6.1. To normalize the token sequences, we use the Lucene analysis infrastructure to convert token sequences to query terms to find entries in the dictionary which is detailed in Sect. 6.2. The dictionary uses the finite-state transducer (FST) implemented in Lucene. We selected it for its query performance and lower memory requirements.

5.2. Link Density

Some mentions are systematically linked in Wikipedia, while others never are. To count the share of matches in linked vs non-linked text, we created a link density measure to assess if a mention should be linked or not:

$$ld(m_i) = \frac{C(m_i \text{ is linked})}{C(m_i)}. \quad (1)$$

We used this metric as a baseline mention filtering method.

We extracted the counts from an auto-linked version of each Wikipedia article. With auto-linking, we refer to the process of linkage densification by using the existing (link, label) pairs per document and creating links for sequences of tokens that exactly match a label.

If multiple links can apply to a same segment, we resolve them using a dominant-right rule, which resolves overlaps into the longest segment or to the furthest to the right for equal length segments.

5.3. Entity, Mentions, and Context

We have two types of statistically extracted mappings:

Monolingual, modeling the relation of mentions and words to entities in one language;

Multilingual, modeling the entity to entities in context over all the editions.

5.3.1. Multilingual

To create the multilingual model, we computed the pointwise mutual (PMI) information of an entity e_i being in the same context as entity e_j:

$$\text{PMI}(e_i, e_j) = \log \frac{P(e_i, e_j)}{P(e_i)P(e_j)}. \quad (2)$$

PMI produces an unbounded value; we therefore used a slightly different formalization to compute the normalized PMI (NPMI) (Bouma, 2009), in this case using raw counts C:

$$\text{NPMI}(e_i, e_j) = \frac{\log \frac{C(e_i, e_j)}{C(e_i)C(e_j)/\text{total}}}{-\log \frac{C(e_i, e_j)}{\text{total}}}, \quad (3)$$

where

$$\text{total} = \sum_i \sum_j C(e_i, e_j). \quad (4)$$

Practically, this is computed by counting the occurrences of two entities in the same context, ultimately resulting in a sorted list of entities based on PMI values. Specifically, there is one list per entity e_i with all other entities e_j found in the same context as e_i. This entity e_i list may be large and is filtered to only retain the top-k e_j entity entries.

The context in this paper is defined as the entities in the same paragraph. Top-k filtering produces sub-optimal results as PMI will favor entities with strong but not statistically significant mutual occurrences. This property is not desirable for frequently occurring entities as semantically relevant entities, in this case, will have lower PMI values and thus be filtered out.

To improve lists of entity-to-entity PMI, a cutoff based on the raw counts is dynamically determined for each list. Specifically, all counts are sorted from highest to lowest counts and the count limit is determined by when a total share of 80% has been reached, or a minimum of 2 if there are too few entries. Subjectively, this heuristic produces better results for frequently occurring entities. It makes little difference for rare entities, where we do not have statistically significant mutual occurrences. Nonetheless, it implies more noise for less frequent entities.

We used information from all nine editions for this computation.

5.3.2. Monolingual

In addition to multilingual entity relations, we computed localized mention/word to entity relations using the same method as above. Words have a minimum NPMI value of 0.1 to surpass, which seems to be where most nonsense words start to appear. This loosely results in a statistical basis for linkage based on words, mentions, and other entities.

6. Setup

6.1. Segmentation and Tokenization

We created a tokenizer using a custom JFlex parser optimizer to find common patterns in many languages. The parser is fully defined in that unknown characters or rules will yield separate tokens. It will split tokens based on whitespace for all relevant languages. The parser also applies rule-based pattern matching to detect acronyms, title-cased words, uppercased words, numbers, years, periods in many Unicode variants, citation, parenthesis, etc.
Sentence splitting is based on rules and uses the tokenization stream and its token classification as input. Concretely, it uses a split rule with a minimum sentence length of four tokens to retain data even if the segmentation is incorrect, but also to avoid common rule-based mistakes: mistakes that frequently occur around the period, such as certain acronyms, person titles and non-standard word contractions. Such mistakes are hard to avoid without the use of a machine-learning model.

6.2. Mention detection

The mention detection consists of two steps:

1. A dictionary-based detection, classification, and overlap resolution using dominant right rule;
2. A neural NER model with dictionary detection as part of the input features.

6.2.1. Baseline: Dictionary Detection

Using only the localized mention dictionary and cherry-picked link density (ld) cutoff values per language[10], we extracted mentions from the input sentences. However, this only finds potential mentions without class information. Named entity classes are determined using candidate generation and picking the top candidate entity and its class according to our entity class mapping from Sect. 4.3.

6.2.2. Named Entity Recognition using ML

The mention detector model shown in Fig. 2 is a BILSTM-BILSTM CHAR-CRF model. Concretely, this model contains the following elements:

- Character sequences per word using a BILSTM layer, essentially compressing embeddings into one vector and concatenated with word embedding;
- Word sequences in sentences using a BILSTM layer, using three inputs: word embedding, character embeddings from above and dictionary features per class from the dictionary detection;

[10] 0.35 for English, 0.25 for Spanish and 0.45 for Chinese

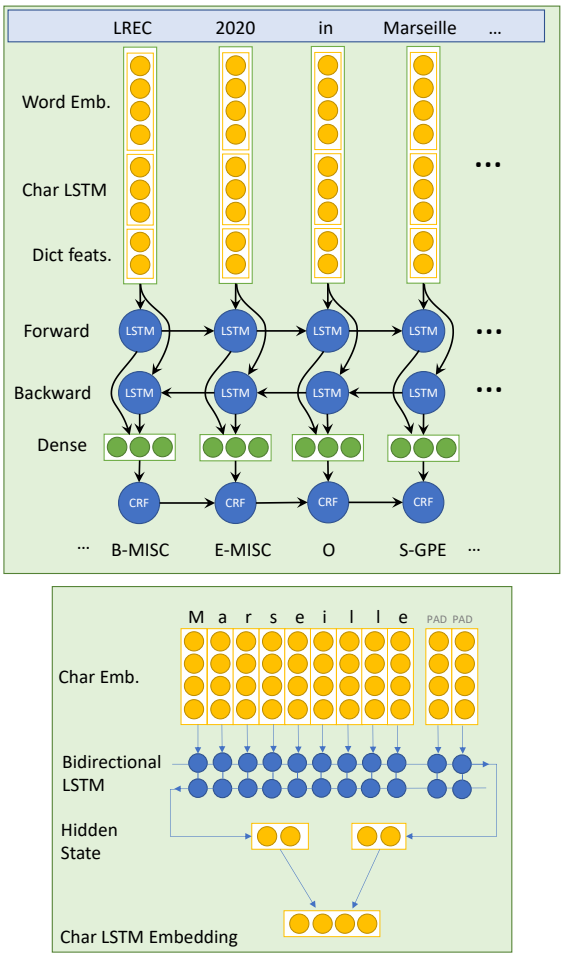

Figure 2: BILSTM-CRF NER model

- Finally, all output is downprojected and fed into a linear chain CRF layer to predict the most likely tag sequence.

This model predicts all mentions used for linkage. To generate candidates, we take token spans as marked by the model and use the mention dictionary to find all the candidates.

Model details: The NER models are trained independently for each language with different hyper parameters, which may affect the architecture. For instance, the Chinese NER models skips the CHAR BILSTM features as characters are treated as words. This makes the combined features for the Chinese NER model to be based on individual characters and the per class features from dictionary detection. The Chinese NER model supports frequently occurring words and acronyms written in other scripts and they will be treated as words not as a sequence of characters.
The notable hyper-parameters we used during training are:

Character features: One (Spanish) or two BILSTM (English) layers, 64 dimensions, hidden state as final feature vector, 25% dropout on this hidden state vector;

Word features: 256 dimension embeddings, 10% word dropout (randomly replacing words with an unknown word embedding);

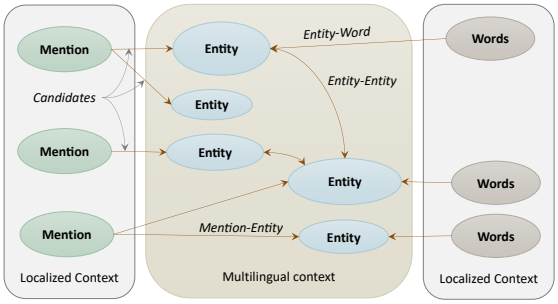

Figure 3: Candidate graph

Combined features: One or two (Chinese) bidirectional LSTM layers, 100 dimensions, 25% dropout. This feature vector is run through a batch normalization layer before being fed to the BILSTM layer;

Epochs: 48 with early stopping based on a validation set evaluation;

Optimizer: Adam with default learning rate and a weight decay of 10^{-4}.

Additionally, within a batch, repeated words will reuse the same character embeddings. The CRF layer and its loss function come from AllenNLP (Gardner et al., 2018).

6.3. Linking Stage

We start with all candidate lists consisting of the possible Q-numbers. The pagerank method will give a weight to each vertex in a directed graph based on the available edges. Figure 3 shows a candidate graph.

In the context of linking, a vertex is either a word, mention, or an entity. Each candidate entity has an associated list of words, mentions and other entities it co-occurs with. We built these lists using the NPMI cutoff method described in Sect. 5.3. We create the edges using these lists by searching the context for the existence of these mentions, words and candidate entities, and generating vertices and directed edges towards all the relevant candidates. The context is unbounded meaning the entire document. This produces a graph with directed edges: from word, mentions to candidate entities, and entity to entity edges.

The final output is a normalized pagerank value for each candidate per mention. The normalized pagerank values are sorted to produce a ranking list. The highest ranking candidate is selected for linkage. There is a minimal NIL detection, filtering out entities which have the NONE class. Our pagerank implementation uses power iteration until convergence or a maximal number of iterations with a default damping value of 0.15 and an α value of 0.85.

7. Results

We evaluated our system with `neleval`[11] and Table 1 shows the resulting scores. Among all the figures output by `neleval`, we included the following ones:

NER: Named entity recognition score;

NERC: NER and type classification (e.g. person, organization, etc.);

KBIDs: Linkage per document;

CEAFmC+: Identification, classification and linking.

The baseline is weak, which was expected as it only uses the mention dictionary. Compared to the results we obtained using a reranker (Klang et al., 2017), the new model is competitive, and improves on almost all metrics except NER on English and linkage on English and Spanish; however the differences are small.

A general observation is that the model favors precision over recall. Chinese is particularly strong on entity linkage and increased the most over the previous model (CEAFmC+). For a full comparison, please refer to the complete results from the TAC 2017 Entity Discovery and linkage overview (Ji et al., 2017).

8. Discussion

8.1. Mention detection

Mention detectors based on neural models are not perfect and can have recall issues. When trained on a noisy dataset such as TAC, they tend to favor precision heavily, resulting in some mentions not being predicted. One way of mitigating this is to expand mentions by using the found mentions and searching for partial or full matches, e.g. linking all occurrences of "Obama" when "Barack Obama" is found as a mention.

TAC contains a large amount of spelling mistakes, particularly in the discussion forum texts. Domain specific variations which produce different surface forms are also commonplace such as e.g. "Microsoft" and "Micro$oft" forms, which are easily recognizable to humans.

When training neural models such as LSTM, it can be tricky to know when to stop. We then employ model checkpointing using the F1 score on a validation set during training, ultimately, picking the final parameters from the best saved model. The validation set in this case is a 10% sample of the original data. Training on all data might yield an improvement. For Spanish and Chinese, improvements might be larger as their corpus sizes are about 50% of English, and in some cases even less when broken down into annotated tags. From the results, the NOM category is the most difficult category to accurately discover and link as it tends to be highly ambiguous. The available training data for nominative mentions (NOM) are sparse with more variation than named entity observations, making learning useful patterns more difficult.

One potential approach that could mitigate the lack of annotated data would be to train on TAC 2016 data, predict on TAC 2015 data and then merge and train again as described by Bernier-Colborne et al. (2017).

We trained the embeddings on relatively limited corpora. The motivation was that these can be reproduced in any language that has a Wikipedia edition, enabling us to expand the linker to many more languages. The disadvantage of it is that their quality is probably inferior to that of publicly available pre-trained embeddings. Another possible improvement would be to use such embeddings that are available for languages in TAC.

[11] `https://github.com/wikilinks/neleval`

		NER			NERC			KBIDs			CEAFmC+		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Tri	Baseline	78.4	43.1	55.6	73.5	40.4	52.1	62.1	53.8	57.6	66.7	36.7	47.3
	Ugglan	89.4	58.4	70.6	83.0	54.3	65.6	80.1	61.7	69.7	70.9	46.4	56.1
	NER-Only	91.5	61.6	73.6	87.9	59.2	**70.8**	0.0	0.0	0.0	12.9	8.7	10.4
	Pagerank	91.4	61.7	**73.7**	86.2	58.3	69.5	82.4	63.8	**71.9**	74.3	50.2	**59.9**
Eng	Baseline	81.1	41.3	54.7	75.4	38.4	50.9	65.3	52.1	58.0	67.3	34.3	45.4
	Ugglan	90.6	65.0	**75.7**	83.8	60.1	70.0	82.3	62.1	**70.7**	69.4	49.8	58.0
	NER-Only	93.2	62.8	75.0	89.2	60.2	**71.9**	0.0	0.0	0.0	22.1	14.9	17.8
	Pagerank	92.5	63.4	75.3	85.6	58.7	69.6	77.4	64.6	70.4	71.6	49.0	**58.2**
Spa	Baseline	71.5	47.7	57.3	64.7	43.2	51.8	52.7	55.1	53.8	55.7	37.2	44.6
	Ugglan	88.5	59.1	70.8	84.8	56.6	67.9	83.9	59.5	**69.6**	70.0	46.7	56.0
	NER-Only	92.2	58.4	**71.5**	88.5	56.0	68.6	0.0	0.0	0.0	21.7	13.7	16.8
	Pagerank	92.2	58.3	**71.5**	88.7	56.0	**68.7**	84.6	58.4	69.1	74.0	46.8	**57.3**
Cmn	Baseline	83.0	41.0	54.9	80.6	39.8	53.3	71.8	53.9	61.5	76.7	37.9	50.7
	Ugglan	89.1	57.4	69.8	82.4	53.1	64.6	85.4	64.3	73.4	71.2	45.8	55.8
	NER-Only	90.0	63.0	**74.1**	86.8	60.8	**71.5**	0.0	0.0	0.0	13.8	9.7	11.4
	Pagerank	90.0	63.0	**74.1**	85.0	59.5	70.0	84.8	68.2	**75.6**	76.4	53.5	**62.9**

Table 1: Results: CEAFmC+ corresponds to the final identification and linking score

8.2. Linker

The linker in Hedwig uses a plain non-personalized variant of PageRank, which is solved using the power iteration algorithm. The fact that it is non-personalized means that important weight information with regards to available computed PMI values might reduce its ability to resolve important entities favoring frequent entities which in this context improves the results.

The linker stage uses the entire document with all possible candidates to form collective agreement; the consequence is that all unique candidates will receive a single weight. This means that the linker is unable to model topic drift or a change of meaning for different mentions. It will pick the global most coherent choice. For TAC, this limitation is reasonable but might not hold for longer texts. In Ugglan, a moving overlapping window was used and the same approach might be applied to Hedwig.

Compared to Ugglan (Klang et al., 2017), which had a machine-learned reranker for the linking component, we can see that for English and Spanish, the results are slightly worse for linkage. A future improvement might be to add a reranker.

9. Acknowledgments

This research was supported by Vetenskapsrådet, the Swedish research council, under the *Det digitaliserade samhället* program, grant number 340-2012-5738.

We are also grateful to the NVIDIA GPU Grant program for providing us with a Titan GPU.

10. Bibliographical References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Bernier-Colborne, G., Barrière, C., and Ménard, P. A. (2017). CRIM's systems for the tri-lingual entity detection and linking task. In *TAC*.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Bunescu, R. and Paşca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Cheng, X. and Roth, D. (2013). Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Cucerzan, S. (2007). 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 Natural Language Learning (EMNLP-CoNLL)*.

Ferragina, P. and Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM.

Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171. Association for Computational Linguistics.

Ganea, O.-E. and Hofmann, T. (2017). Deep joint en-

tity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Ganea, O.-E., Ganea, M., Lucchi, A., Eickhoff, C., and Hofmann, T. (2016). Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July. Association for Computational Linguistics.

Globerson, A., Lazic, N., Chakrabarti, S., Subramanya, A., Ringaard, M., and Pereira, F. (2016). Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 621–631.

Guo, Z. and Barbosa, D. (2014). Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 499–508. ACM.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Ji, H., Pan, X., Zhang, B., Nothman, J., Mayfield, J., McNamee, P., and Costello, C. (2017). Overview of TAC-KBP2017 13 languages entity discovery and linking. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST.

Jiang, Y., Hu, C., Xiao, T., Zhang, C., and Zhu, J. (2019). Improved differentiable architecture search for language modeling and named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3585–3590, Hong Kong, China, November. Association for Computational Linguistics.

Klang, M. and Nugues, P. (2018). Linking, searching, and visualizing entities in wikipedia. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3426–3432.

Klang, M. and Nugues, P. (2019). Docria: Processing and storing linguistic data with Wikipedia. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Turku, October.

Klang, M., Dib, F., and Nugues, P. (2017). Overview of the Ugglan entity discovery and linking system. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Straková, J., Straka, M., and Hajic, J. (2019). Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy, July. Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Zhang, S., Jiang, H., Xu, M., Hou, J., and Dai, L. (2015). The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 495–500.