A Multilingual Entity Linker Using PageRank and Semantic Graphs

Anton Södergren
Department of computer science
Lund University, Lund
karl.a.j.sodergren@gmail.com

Pierre Nugues
Department of computer science
Lund University, Lund
pierre.nugues@cs.lth.se

Abstract

This paper describes HERD, a multilingual named entity recognizer and linker. HERD is based on the links in Wikipedia to resolve mappings between the entities and their different names, and Wikidata as a language-agnostic reference of entity identifiers.

HERD extracts the mentions from text using a string matching engine and links them to entities with a combination of rules, PageRank, and feature vectors based on the Wikipedia categories. We evaluated HERD with the evaluation protocol of ERD'14 (Carmel et al., 2014) and we reached the competitive F1-score of 0.746 on the development set. HERD is designed to be multilingual and has versions in English, French, and Swedish.

1 Introduction

Named entity recognition (NER) refers to the process of finding mentions of persons, locations, and organizations in text, while entity linking (or disambiguation) associates these mentions with unique identifiers. Figure 1 shows an example of entity linking with the mention *Michael Jackson*, an ambiguous name that may refer to thousands of people and where 21 are famous enough to have a Wikipedia page (Wikipedia, 2016). In Fig. 1, the search engine selected the most popular entity (top) and used the cue word *footballer* (bottom) to link the phrase *Michael Jackson footballer* to the English defender born in 1973.

Entity recognition and linking has become a crucial component to many language processing applications: Search engines (Singhal, 2012), question answering (Ferrucci, 2012), or dialogue agents. This importance is reflected by a growing number of available systems; see TAC-KBP2015 (Ji et al., 2015), for instance, with 10 participating teams.

Although many applications include entity linkers, the diversity of the input texts, which can include tweets, search queries, news wires, or encyclopedic articles, makes their evaluation problematic. While some evaluations consider entity linking in isolation and mark the mentions in the input, end-to-end pipelines, where the input consists of raw text, need to combine entity recognition and linking. The ERD'14 challenge (Carmel et al., 2014) is an example of the latter.

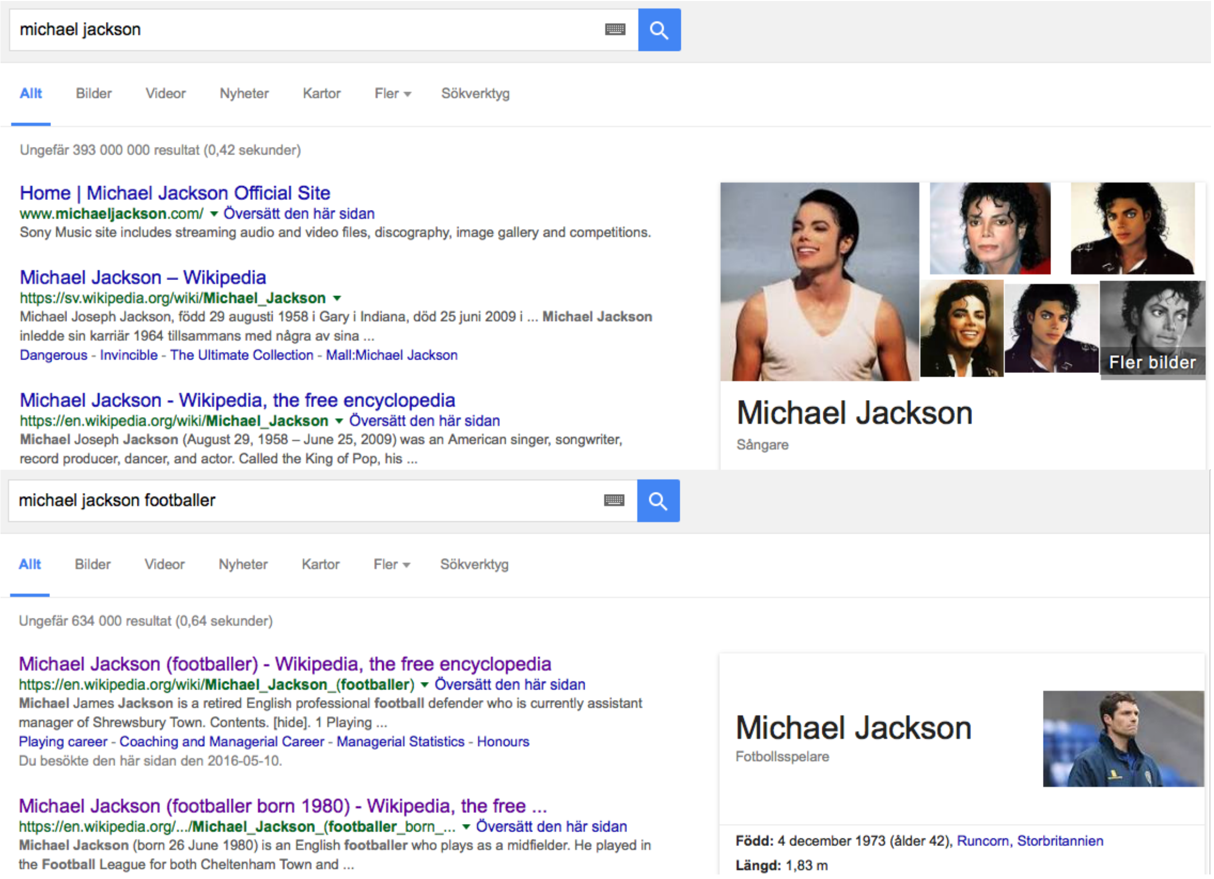
2 Previous Work

Entity linking has spurred a considerable amount of work over the last 10 years. Bunescu and Pasca (2006), Mihalcea and Csomai (2007), and Cucerzan (2007) used Wikipedia as a knowledge source and its articles to define the entities, its hyperlinks to find the mentions, and semantic knowledge from redirect pages and categories, to carry out disambiguation. Milne and Witten (2008) used the likelihood of an entity given a mention M, $P(E|M)$, and a relatedness metric between two entities computed from the links to their corresponding pages to improve both recall and precision. Ferragina and Scaiella (2010) addressed shorter pieces of text with the idea to use a collective agreement between all the entities.

The Entity Recognition and Disambiguation Challenge (ERD'14) (Carmel et al., 2014) is a recent evaluation, where competitors were given a set of entities to recognize and link in a corpus of unlabelled text. This setting is closer to real world application than TAC (Ji et al., 2015), where participants have to link already bracketed mentions. The evaluation included two tracks: one with long documents of an average size of 600 words and a short track consisting of search query.

Finally, the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003) is an influen-

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 87–95,
Gothenburg, Sweden, 23-24 May 2017. ©2017 Linköping University Electronic Press



michael jackson

Ungefär 393 000 000 resultat (0,42 sekunder)

Home | Michael Jackson Official Site
www.michaeljackson.com/ ▾ Översätt den här sidan
Sony Music site includes streaming audio and video files, discography, image gallery and competitions.

Michael Jackson – Wikipedia
https://sv.wikipedia.org/wiki/**Michael_Jackson** ▾
Michael Joseph Jackson, född 29 augusti 1958 i Gary i Indiana, död 25 juni 2009 i ... **Michael Jackson** inledde sin karriär 1964 tillsammans med några av sina ...
Dangerous - Invincible - The Ultimate Collection - Mall:Michael Jackson

Michael Jackson - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Michael_Jackson** ▾ Översätt den här sidan
Michael Joseph **Jackson** (August 29, 1958 – June 25, 2009) was an American singer, songwriter, record producer, dancer, and actor. Called the King of Pop, his ...

Fler bilder

Michael Jackson
Sångare

michael jackson footballer

Ungefär 634 000 resultat (0,64 sekunder)

Michael Jackson (footballer) - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Michael_Jackson_(footballer)** ▾ Översätt den här sidan
Michael James **Jackson** is a retired English professional **football** defender who is currently assistant manager of Shrewsbury Town. Contents. [hide]. 1 Playing ...
Playing career - Coaching and Managerial Career - Managerial Statistics - Honours
Du besökte den här sidan den 2016-05-10.

Michael Jackson (footballer born 1980) - Wikipedia, the free ...
https://en.wikipedia.org/.../**Michael_Jackson_(footballer**_born_... ▾ Översätt den här sidan
Michael Jackson (born 26 June 1980) is an English **footballer** who plays as a midfielder. He played in the **Football** League for both Cheltenham Town and ...

Michael Jackson
Fotbollsspelare

Född: 4 december 1973 (ålder 42), Runcorn, Storbritannien
Längd: 1,83 m

Figure 1: Search results for the queries *Michael Jackson* and *Michael Jackson footballer*. The engine returns the most popular entity (top) and uses the minimal context given in the query, *footballer*, to propose a less popular entity (bottom)

tial evaluation of language independent named entity recognition, with a focus on German and English. Hoffart et al. (2011) linked the names in the English corpus to Wikipedia pages making this dataset a useful corpus for entity linking.

In this paper, we used the ERD'14 development set as well as the CoNLL-2003 dataset with Wikidata links.

3 Building the Entity Knowledge Base

3.1 Mention-Entity Pairs

Following Mihalcea and Csomai (2007) and Bunescu and Pasca (2006), we collected the mention-entity pairs from the Wikipedia links (wikilinks). We built the entity base from the from three versions of Wikipedia: English, French, and Swedish, and the frequency of the wikilinks in each version. Figure 2 shows an example of a pair with the mention *Swedish Parliament*. This gives suggestions of how an entity is commonly referred to: i.e. its name or aliases.

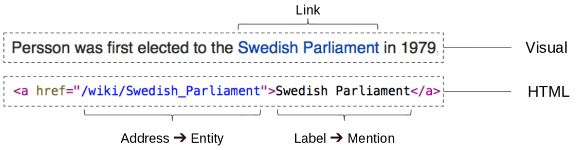

Link
Persson was first elected to the Swedish Parliament in 1979
Visual
<a href="/wiki/Swedish_Parliament">Swedish Parliament</a>
HTML
Address → Entity
Label → Mention

Figure 2: The structure of wikilinks in Wikipedia.

3.2 Entity Nomenclature

As nomenclature for the entities, we used Wikidata, the linked database of Wikimedia. Wikidata connects the different language versions of each article in Wikipedia with a unique identifier called the Q-number. In addition to being a cross-lingual repository, Wikidata also links a large number of entities to structured information such as the dates of birth and death for persons.

In order to use a language-agnostic identifier, we translated the wikilinks into Q-numbers. We extracted the pairs of Q-numbers and article names from a Wikidata dump for each language. Since the dump does not contain any URL, the algorithm must recreate the address from the titles. We could reach a coverage of about 90%. The remaining 10% corresponds to *redirect pages* that act as alternative names or to cover common misspellings. We used the Wikipedia dump to identify these redirects and we improved the coverage rate to 99.1%.

3.3 Annotating the Mentions

We annotated the mention-entity pairs in our knowledge base with a set of features that we used in the subsequent processing:

1. The `Frequency` of the mention-entity pairs. Table 1 shows an example for the city of Honolulu.

2. We used dictionaries of common nouns, verb, and adjectives for each language. If a mention only consists of words in the dictionary, we mark it as `only-dictionary`. An example of this is the artist *Prince* in English and French.

3. We computed a list of the most frequent words (stop words) from Wikipedia for each language. They include *the*, *in*, *and*, and *a*, in English. If all the words in a mention are stop words, we mark it as `only-stop-words`.

4. The system marks the mentions with a high number of links as `highly-ambiguous`, such as *John* or *details* with almost 5,000 different entities linked to each.

5. Mentions without uppercase letters are marked as `lower-case`.

6. Family names and surnames. If the most common mention of a person has more than two words, we mark each mention of one word as `generic`, such as the mention *Bush* referring to the former president George W. Bush.

7. We also annotate the entities with their frequency (`total-frequency`). It corresponds to the sum of all their mentions frequencies.

3.4 Pruning the Knowledge Base

Although Wikipedia is reviewed by scores of volunteers, there are plenty of misleading mentions in the collected knowledge base. We removed a part of them using the following rules:

1. The mention is marked as `lower-case` and either `only-dictionary` or `only-stop-words`

2. The `frequency` of the entity-mention is exactly 1, while the `total-frequency` of that entity is above a threshold parameter that was empirically obtained.

3. The mention consists of two or more words, and starts with a lower-case stop word, that is not a definite article. This will filter out anchors of the type *a city in Sweden*.

We also clustered the mentions with a normalized Levenshtein distance (Levenshtein, 1966). If the distance was less than 0.2 to any element, they were considered to be in the same group. We applied the clustering to all the surface forms and we discarded the mentions without a group.

4 System Components and Pipeline

The system consists of three main components: a spotter that identifies the mentions, a set of rules that prunes the results, and an improver that uses contextual clues for entity recognition (Fig. 3).

The spotter outputs a match for every conceivable mention of an entity, leaving us with a document where almost every word is tagged as an entity. This output has a very high recall, but a very low precision.

The filtering that comes after that tries to remove the most unlikely of the alternatives from the spotter and raises the precision to a modest level while trying to have a minimal impact on the recall. If the input data to the next step is completely unfiltered the Contextual Improver is unable to affect the results in a positive way.

The Contextual Improver is the final step and uses contextual clues, such as which categories the entities belong to and which entities are commonly seen together, to improve the result.

4.1 Mention Spotting

We use the mention-entity knowledge base to spot the mentions in raw text and associate them with all their possible entities. Following Lipczak et al. (2014), we applied the Solr Text Tagger (Smiley, 2013) based on finite-state transducers and the Lucene indexer. Solr Text Tagger was chosen as it is a highly effective way of marking up possible matches of a database in a text. It is based on the

Mention	Entity	Frequency	Mention	Entity	Frequency
Honolulu	Q18094	5117	Hawaii	Q18094	11
Honolulu, Hawaii	Q18094	2281	Honolulu, HI MSA	Q18094	7
Honolulu, HI	Q18094	600	HONOLULU	Q18094	7
Honolulu, Hawaii, USA	Q18094	67	city of Honolulu	Q18094	7
Honolulu, Hawai’i	Q18094	47	honolulu	Q18094	5
Honolulu, Hawai'i	Q18094	21	Honululu	Q18094	5
Honolulu CPD	Q18094	21			

Table 1: An example of the mention-entity counts for the entity Q18094, most commonly called Honolulu

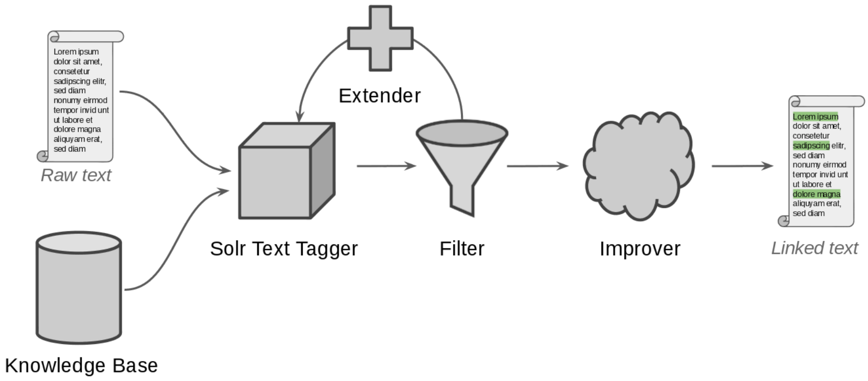

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
Raw text
Extender
Solr Text Tagger
Filter
Improver
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
Linked text
Knowledge Base

Figure 3: The system architecture, where the knowledge base contains the mention-entity pairs

Lucene open-source software and its implementation of finite-state transducers.

As a preprocessing step, Solr Text Tagger compiles all the mentions in the knowledge base, where the input labels are the letters and symbols of mentions and the output labels are the entity identifiers. Then, given an untagged text, the tagger marks all the occurrences, possibly overlapping, of all the names in the database. See Fig. 4.

4.2 Filtering and Expansion

The output of the Spotter is usually very noisy as most words can match some mention in the knowledge base. Examples include *It*, a novel by Stephen King, or *Is This It*, an album by The Strokes. The result is a high mention recall, but a low precision. We applied filters to remove some matches and improve the precision while preserving the recall.

The system uses a set of manually-written rules and empirically obtained hyper parameters to improve precision with a minimal effect on recall. We describe them in the sections below. The complete list of parameter values is provided in the HERD source code available from GitHub[1].

4.2.1 Mention Probability

We computed the probability for a term or a phrase to be a link over the whole Wikipedia (Eckhardt et al., 2014) using the formula: Mention probability $= \frac{link(mention)}{freq(mention)}$. This gives a hint at whether a word sequence is more commonly used as a mention of entities or just as words.

Medical Center, for example, is linked 1.0% of the times used, while *Medical Center of Central Georgia* has a mention probability of 73.7%.

Any candidate that had less than 0.5% mention probability was immediately pruned.

4.2.2 Filters to Improve Precision

Filters are rules based on syntactical clues and the flags defined in Sect. 3.3. Each matching rule returns a suspicion score and we compute the sum of the scores. The most significant rules are:

1. Capitalization: Add suspicion to any mention that does not contain a capital letter. This loses some recall, but increases the precision

[1] `https://github.com/AAAton/herd`

dramatically. We improved it with a function that takes into consideration the number of capital letters, whether the mention is the start of a new sentence and whether the mention has the `only-dictionary` tag. Headlines also use capitalized words. We recognize fully capitalized sentences with regular expressions. Mentions in headlines with the `only-dictionary`, or `only-stop-words` tags, generate suspicion.

2. Generic names: We apply a two-pass analysis for `generic` mentions. We remove them from the first pass. In a second pass, the generic names are restored if a mention of the same entity that is not generic shows in the text i.e. the *Bush* mention is kept only when there is a full mention of *George W. Bush* in the text. This is to avoid the tagging of the mention *Bush* with every entity having this name as a generic name.

3. Colliding names: If two tags are immediate neighbors, with the exception of a space, they generate suspicion. The system considers all the candidate entities for a surface form, and only triggers suspicion if at least one of the entities is a generic name. This is a method to detect and avoid tagging a multiword name which does not exist in our knowledge base with multiple other candidates.

4.2.3 Extender to Improve Recall

We extended the detected mentions following the work of Eckhardt et al. (2014). Once we recognize a mention consisting of two words or more that passed the filter, we create new mentions with acronyms and generic version of the item by splitting it into multiple parts.

Given the text *a division of First Citizens BancShares Inc. of Raleigh, N.C.*, where the system recognizes the mention *First Citizens BancShares Inc*, the extender creates possible acronyms, such as *FCBI* and *F.C.B.I.* It also looks for parentheses, immediately following a mention, giving a suggestion of how it is meant to be abbreviated.

The extender also splits the mention into parts of 1, 2, and 3 words. The mention above generates *First*, *Citizens*, *BankShares* and *Inc.*, as well as *First Citizens*, *Citizens BankShares*, *BankShares Inc*, and so forth. We associate the generated mentions with the set of entities of the original mention. The tagged extensions are filtered in the same manner as all the other tags.

4.3 Contextual Improver

For each document, the Improver uses PageRank to determine which candidates commonly occur together, and prunes the unseen combinations. This produces a result with a very high precision. We use this high precision output as the modelled context of the document. A weighted category graph is calculated for this modelled context. The named entity recognition is then widened by considering the similarity of all the candidates to the modelled context.

Once the improver has been applied, we carry out a final clean-up step, where we eliminate all the overlap and rank the remaining candidates for each mention.

4.3.1 PageRank

We apply a modified version of PageRank (Brin and Page, 1998) to the tagged mentions. Following Eckhardt et al. (2014), we create a node for every mention-entity pair that is detected in the text, and run three iterations of PageRank. We analyze the internal links of Wikipedia to determine which entities appear in the same context. Two entities are considered linked if the article of Entity *A* links to the article of Entity *B* or a link to both the article of Entity *A* and the article of Entity *B* occurs in the same paragraph.

The links calculated on Wikipedia are transferred to the tagged document and produce a graph of linked entities. Unlike the work of Eckhardt et al. (2014), we initialize each node with the standard value $1/N$, and the ratio between the initial value and the final value is used to modify the suspicion for a candidate.

After applying PageRank, some entities will be higher ranked than others which we take as an input to another round of filtering. The candidates with high suspicion are removed. This produces a high precision output, with a slight drop of the recall.

4.3.2 Weighted Category Graph

We use a weighted category graph (WCG) derived from the user-annotated categories of Wikipedia articles (Lipczak et al., 2014). For each article, we created this graph from all the languages in which the article is available. If an article has the same category in all the languages, this category is as-

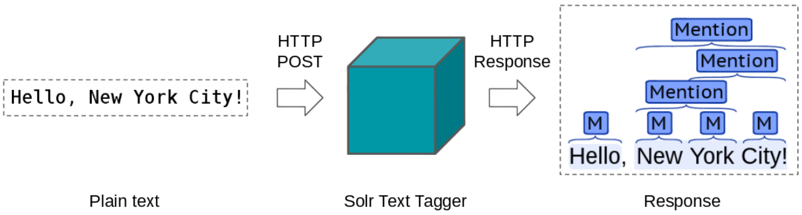


Figure 4: The Solr Text Tagger processing scheme, where the operations are carried out through POST requests.

signed the maximum weight of 1.0. The weight is then decreased as a linear function of the number of languages. The categories need to be weighted to avoid rare or incorrect categories assigned to articles.

Wikipedia categories are themselves categorized into parent categories. The category of *Sweden* has a parent category of *Countries in Europe* for example. A tree of categories is derived from each article by getting the top *k* categories of an article, and expanding those categories with the top *k* parents for each category. This process is repeated *d* times. This generalizes the categories of the article, which makes them easier to compare to adjacent articles.

The value *k* is set to 5, and the value *d* is set to 3, as proposed by Lipczak et al. (2014).

Table 2 shows an example of the categories we obtain from the article about *Sweden*.

The weighted category graph is used in the following way:

1. We input a set of core entities with high precision. The improver calculates a weighted category vector for each entity and creates a `topic centroid` as the linear combination of these vectors. This is meant to function as the general topic of the analyzed document.

2. We improve the precision of the core entities by comparing each of the high-precision entities to the topic centroid with a cosine similarity. If the score of an entity is under a threshold value of 0.6, it is removed. Finally, the topic centroid is recalculated.

3. We then compare each entity in the unfiltered output of Solr Text Tagger to the topic centroid with a cosine similarity. We keep the entities that are above a threshold value of 0.2.

Category	Ratio
Sweden	1.00
Countries in Europe	0.38
Member states of the European Union	0.30
Constitutional monarchies	0.20
Scandinavia	0.20

Table 2: Top 5 categories for Sweden, Q34

This procedure widens the scope of the entity selection and improves the recall in a context aware manner. Since the output of the weighted category graphs is similar to the input constraints of PageRank, and the output of PageRank is similar to the input requirements of the weighted category graph, we have set them into an iteration cycle in order to achieve higher results.

4.3.3 Clean-up

As a final step, we eliminate overlapping mentions: When two mentions overlap, we keep the longest. If the mentions are of equal length, we keep the rightmost.

The remaining candidates for each mention are then ranked by their wikilink frequency, and the most frequent candidate is selected as the correct disambiguation.

5 Experimental Setup and Results

We evaluated the system with two different data sets for English: ERD-51 and AIDA/YAGO and we used the evaluation metrics of ERD'14 Carmel et al. (2014). We did not have access to evaluation sets for the other languages.

ERD-51 is the development set of Carmel et al. (2014). It consists of 51 documents that have been scraped from a variety of sources with 1,169 human-annotated mentions. Each annotation has a start, an end, and a Freebase identifier (Bollacker et al., 2008). In the competition, a set of entities, slightly over 2 million, was given, and thus the problem of defining what a named entity actually is was avoided. We filtered our knowledge base to only contain mentions of entities from the given set.

AIDA/YAGO is derived from the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). It is a collection of 1393 news articles with 34,587 human-annotated names. 27,632 of those names have been disambiguated with a link to the corresponding Wikipedia site using a dump from 2010-08-17 (Hoffart et al., 2011).

Tables 4 and 3 show the results evolving over different versions of the system. The execution time is normalized to the time spent to process 5,000 characters. It should not be considered an absolute value, but can be used to compare the different algorithms.

Lipczak et al. (2014) and Eckhardt et al. (2014), respectively second and third of ERD'14, reported results of 0.735 and 0.72 on the test set. Our results are not exactly comparable as we used the development set. Nonetheless we believe our system competitive. The highest scoring participant (Cucerzan, 2014) with a score of 0.76 was one of the organizers and was not part of the competition. The reason for the difference in recognition score between the ERD-51 and the AIDA/YAGO dataset lies in the text genres. AIDA/YAGO is collected from well written news wires, mainly written by professionals with proper punctuation and capitalization, while ERD-51 is a collection of more poorly written texts collected from a wider variety of sources, spanning from blogs to eBay sales sites.

For the score of the linked systems, the results are the opposite. ERD-51 has been annotated and linked by humans from a predefined set of 2 million entities, while the linking in AIDA/YAGO has been done from an older dump of Wikipedia. The dump contains around 3 million different sites, unlike the dump we used that has around 5 million entities. Both the mismatch in the entities that are possible to use and the larger size of the knowledge base lead to a larger gap between recognition and linking.

With a latency of around 200 ms per 5,000 characters, the system should be able to tag the entire of the English Wikipedia under a day with a heavy duty computer. We estimate that the average length of a Wikipedia article is around 2,000 characters, that there are 5 million articles, 24 available cores and 100% run time.

We implemented the entity linker as an interactive demonstration. The user can paste a text and visualize the results in the form of a text annotated with entity labels. Once the user hovers over the label, the ranked candidates are displayed with a link to the Wikidata page for that entity. Figure 5 shows an output of the system.

6 Conclusions and Future Work

We explored different methods to carry out language-independent entity linking from raw text and we presented evaluations on English. The version of the system that had the highest score used a 4-step pipeline, ending with an iteration cycle between a personalized version of PageRank and the usage of weighted category graphs. The system reached a weighted F1-score of 0.746 on the ERD-51 dataset.

The paper takes an unusual approach to named entity recognition and disambiguation as it does not separate the tasks, but treats every candidate to every mention as a separate possibility. The iteration between two context-aware algorithms with different precision/recall characteristics improved the results dramatically and is, to the best of our knowledge, a novel, language-independent approach to entity recognition and disambiguation. We also exposed a way of pruning unsatisfying links in a collected knowledge base by clustering.

The system and its approach can be improved in a number of ways. Firstly, the usage of manually written rules is suboptimal, as the rules may be dependent on the language. A method that only uses Wikipedia, or some other multilingual resource, and avoids syntactical clues altogether, would be a better solution. An approach like that would be more robust to the differences between languages.

We have also introduced a number of threshold values in different parts of the system, such as the suspicion cutoff in the manual rules and the similarity threshold in the WCG. These are difficult to optimize without an automated method. We think HERD would benefit from finding an alternative approach that avoids cutoff values or automates their optimization.

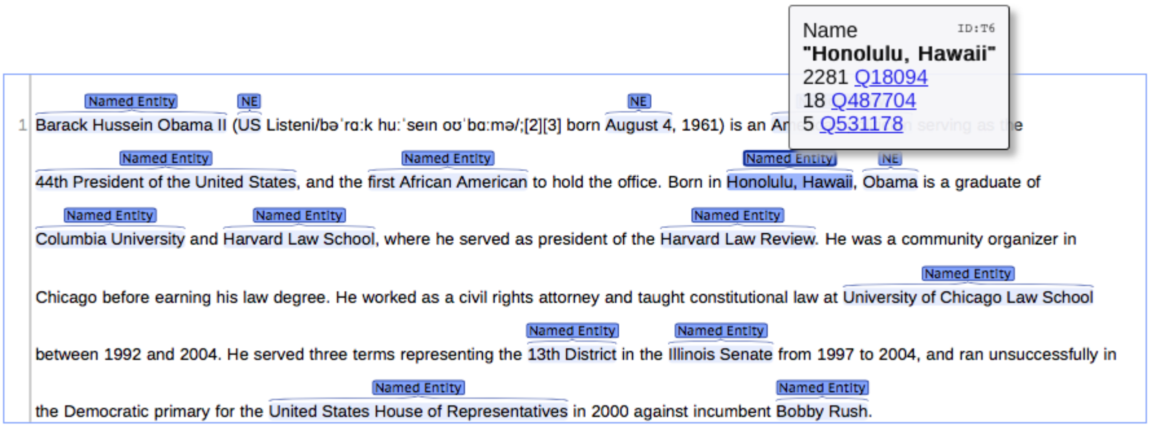
The approach would also benefit from a deeper evaluation on a wider variety of datasets, spanning over several languages, as well as the two English datasets used in this article.

Version	Exec. (ms)	Recognition Precision	Recognition Recall	Recognition F	Linking Precision	Linking Recall	Linking F
v0.1 Baseline	-	0.745	0.686	0.714	0.551	0.521	0.535
v0.2 With extender	-	0.673	0.762	0.715	0.520	0.581	0.549
v0.3 Remove colliding mentions	-	0.698	0.730	0.714	0.536	0.572	0.554
v0.4 Simplified filtering	-	0.723	0.720	0.721	0.613	0.610	0.611
v0.5 Relaxed filtering	-	0.642	**0.859**	0.734	0.522	0.720	0.605
v0.6 Part of headline	-	0.691	0.796	0.740	0.593	0.684	0.635
v0.7 Filter out candidates with low frequency	250	0.718	0.785	0.750	0.612	0.672	0.640
v0.8 PageRank	246	**0.865**	0.765	0.812	**0.747**	0.661	0.701
v1.0 PageRank & WCG	608	0.843	0.848	**0.845**	0.744	0.749	**0.746**

Table 3: Results on the ERD51 dataset

Version	Exec. (ms)	Recognition Precision	Recognition Recall	Recognition F	Linking Precision	Linking Recall	Linking F
v0.1 Baseline	-	0.940	0.630	0.755			
v0.2 With extender	-	0.889	0.684	0.773			
v0.3 Remove colliding nodes	-	0.923	0.678	0.782			
v0.4 Simplified filtering	-	**0.934**	0.705	0.803			
v0.5 Relaxed filtering	-	0.912	0.769	0.835			
v0.7 Filtered out candidates with low frequency	1088	0.888	0.789	0.835			
v0.8 PageRank	3325	0.935	0.819	0.873	0.732	0.631	0.678
v1.0 PageRank & WCG	4415	0.932	0.827	0.876	**0.740**	**0.647**	**0.690**

Table 4: Results on the AIDA/YAGO dataset



Name ID:T6
"Honolulu, Hawaii"
2281 Q18094
18 Q487704
5 Q531178

Barack Hussein Obama II (US Listeni/bəˈrɑːk huːˈseɪn oʊˈbɑːmə/;[2][3] born August 4, 1961) is an Ar
44th President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of
Columbia University and Harvard Law School, where he served as president of the Harvard Law Review. He was a community organizer in
Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at University of Chicago Law School
between 1992 and 2004. He served three terms representing the 13th District in the Illinois Senate from 1997 to 2004, and ran unsuccessfully in
the Democratic primary for the United States House of Representatives in 2000 against incumbent Bobby Rush.

Figure 5: Visualizer

The HERD source code is available from GitHub at this repository: `https://github.com/AAAton/herd`. A demonstration of HERD as part of the Langforia processing pipelines (Klang and Nugues, 2016) is available at this location: `http://vilde.cs.lth.se:9000`.

Acknowledgements

This research was supported by Vetenskapsrådet, the Swedish research council, under the *Det digitaliserade samhället* program.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. 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. ACM.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April.

Razvan C Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *European Chapter of the Association for Computational Linguistics*, volume 6, pages 9–16.

David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. 2014. ERD'14: Entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, volume 48, pages 63–77. ACM.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 7, pages 708–716.

Silviu Cucerzan. 2014. Name Entities Made Obvious: The Participation in the ERD 2014 Evaluation. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 95–100, New York, NY, USA. ACM.

Alan Eckhardt, Juraj Hreško, Jan Procházka, and Otakar Smri;. 2014. Entity linking based on the co-occurrence graph and entity probability. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 37–44, New York, NY, USA. ACM.

Paolo Ferragina and Ugo Scaiella. 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*, CIKM '10, pages 1625–1628, New York, NY, USA. ACM.

David Angelo Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1 –1:15, May-June.

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

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 trilingual entity discovery and linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015).*

Marcus Klang and Pierre Nugues. 2016. Langforia: Language pipelines for annotating large collections of documents. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 74–78, Osaka, Japan, December. The COLING 2016 Organizing Committee.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February.

Marek Lipczak, Arash Koushkestani, and Evangelos Milios. 2014. Tulip: Lightweight entity recognition and disambiguation using wikipedia-based topic centroids. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 31–36, New York, NY, USA. ACM.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA. ACM.

Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. Official Google Blog. `http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`. Retrieved 7 November 2013, May.

David Smiley. 2013. Solr text tagger, text tagging with finite state transducers. `https://github.com/OpenSextant/SolrTextTagger`.

Erik F. Tjong Kim Sang and Fien De Meulder. 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*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wikipedia. 2016. Michael Jackson (disambiguation) – Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Michael_Jackson_(disambiguation)`.