AIDArabic
A Named-Entity Disambiguation Framework for Arabic Text

Mohamed Amir Yosef, Marc Spaniol, Gerhard Weikum
Max-Planck-Institut für Informatik, Saarbrücken, Germany
{mamir|mspaniol|weikum}@mpi-inf.mpg.de

Abstract

There has been recently a great progress in the field of automatically generated knowledge bases and corresponding disambiguation systems that are capable of mapping text mentions onto canonical entities. Efforts like the before mentioned have enabled researchers and analysts from various disciplines to semantically "understand" contents. However, most of the approaches have been specifically designed for the English language and - in particular - support for Arabic is still in its infancy. Since the amount of Arabic Web contents (e.g. in social media) has been increasing dramatically over the last years, we see a great potential for endeavors that support an entity-level analytics of these data. To this end, we have developed a framework called AIDArabic that extends the existing AIDA system by additional components that allow the disambiguation of Arabic texts based on an automatically generated knowledge base distilled from Wikipedia. Even further, we overcome the still existing sparsity of the Arabic Wikipedia by exploiting the interwiki links between Arabic and English contents in Wikipedia, thus, enriching the entity catalog as well as disambiguation context.

1 Introduction

1.1 Motivation

Internet data including news articles and web pages, contain mentions of named-entities such as people, places, organizations, etc. While in many cases the intended meanings of the mentions is obvious (and unique), in many others, the mentions are ambiguous and have many different possible meanings. Therefore, Named-Entity Disambiguation (NED) is essential for many application in the domain of Information Retrieval (such as information extraction). It also enables producing more useful and accurate analytics. The problem has been exhaustively studied in the literature. The essence of all NED techniques is using background information extracted from various sources (e.g. Wikipedia), and use such information to know the correct/intended meaning of the mention.

The Arabic content is enormously growing on the Internet, nevertheless, background ground information is clearly lacking behind other languages such as English. Consider Wikipedia for example, while the English Wikipedia contains more than 4.5 million articles, the Arabic version contains less than 0.3 million ones [1]. As a result, and up to our knowledge, there is no serious work that has been done in the area of performing NED for Arabic input text.

1.2 Problem statement

NED is the problem of mapping ambiguous names of entities (mentions) to canonical entities registered in an entity catalog (knowledgebase) such as Freebase (www.freebase.com), DBpedia (Auer et al., 2007), or Yago (Hoffart et al., 2013). For example, given the text "I like to visit Sheikh Zayed. Despite being close to Cairo, it is known to be a quiet district", or in Arabic,"أحب زيارة الشيخ زايد. فهي تتميز بالهدوء بالرغم من قربها من القاهرة". When processing this text automatically, we need to be able to tell that Sheikh Zayed denotes the the city in Egypt[2], not the mosque in Abu Dhabi[3] or the President of the United Arab

[1] as of July 2014
[2] http://en.wikipedia.org/wiki/Sheikh_Zayed_City
http://ar.wikipedia.org/wiki/مدينة_الشيخ_زايد
[3] http://en.wikipedia.org/wiki/Sheikh_Zayed_Mosque
http://ar.wikipedia.org/wiki/جامع_الشيخ_زايد

Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP), pages 187–195,
October 25, 2014, Doha, Qatar.

Emirates[4]. In order to automatically establish such mappings, the machine needs to be aware of the characteristic description of each entity, and try to find the most suitable one given the input context. In our example, knowing that the input text mentioned the city of Cairo favors the Egyptian city over the mosque in Abu Dhabi, for example. In principle, state-of-the-art NED frameworks require main four ingredients to solve this problem:

- **Entity Repository**: A predefined universal catalog of all entities known to the NED framework. In other words, each mention in the input text must be mapped to an entity in the repository, or to null indicating the correct entity is not included in the repository.
- **Name-Entity Dictionary**: It is a many-to-many relation between possible mentions and the entities in the repository. It connects an entity with different possible mentions that might be used to refer to this entity, as well as connecting a mention with all potential candidate entity it might denote.
- **Entity-Descriptions**: It keeps per entity a bag of characteristic keywords or keyphrases that distinguishes an entity from another. In addition, they come with scoring scheme that signify the specificity of such keyword to that entity.
- **Entity-Entity Relatedness Model**: For coherent text, the entities that are used for mapping all the mentions in the input text, should be semantically related. For that reason, an entity-entity relatedness model is required to asses the coherence.

For the English language, all of the ingredients mentioned above are richly available. For instance, the English Wikipedia is a comprehensive up-to-date resource. Many NED systems use Wikipedia as their entity repository. Furthermore, many knowledge bases are extracted from Wikipedia as well. When trying to apply the existing NED approaches on the Arabic text, we face the following challenges:

- **Entity Repository**: There is no such comprehensive entity catalog. Arabic Wikipedia is an order of magnitude smaller than the English one. In addition, many entities in the Arabic Wikipedia are specific to the Arabic culture with no corresponding English counterpart. As a consequence, even many prominent entities are missing from the Arabic Wikipedia.
- **Name-Entity Dictionary**: Most of the name-entity dictionary entries originate from manual input (e.g. anchor links). Like outlined before, Arabic Wikipedia has fewer resources to extract name-entity mappings, caused by the lack of entities and lack of manual input.
- **Entity-Descriptions**: As already mentioned, there is a scarcity of anchor links in the Arabic Wikipedia. Further, the categorization system of entities is insufficient, Both are essential sources of building the entities descriptions. Hence, it is more challenging to produce comprehensive description of each entity.
- **Entity-Entity Relatedness Model**: Relatedness estimation among entities is usually computed using the overlap in the entities description and/or link structure of Wikipedia. Due to the previously mentioned scarcity of contents in the Arabic Wikipedia, it is also difficult to accurately estimate the entity-entity relatedness.

As a consequence, the main challenge in performing NED on Arabic text is the lack of a comprehensive entity catalog together with rich descriptions of each entity. We considered our open source AIDA system[5] (Hoffart et al., 2011)- mentioned as state-of-the-art NED System by (Ferrucci, 2012) - as a starting point and modified its data acquisition pipeline in order to generate a schema suitable for performing NED on Arabic text.

1.3 Contribution

We developed an approach to exploit and fuse cross-lingual evidences to enrich the background information we have about entities in Arabic to build a comprehensive entity catalog together with their context that is not restricted to the Arabic Wikipedia. Our contributions can be summarized in the following points:

- **Entity Repository**: We switched to YAGO3(Mahdisoltani et al., 2014), the

[4]http://en.wikipedia.org/wiki/Zayed_bin_Sultan_Al_Nahyan
http://ar.wikipedia.org/wiki/زايد_بن_سلطان_آل_نهيان

[5]https://www.github.com/yago-naga/aida

multilingual version of YAGO2s. YAGO3 comes with a more comprehensive catalog that covers entities from different languages (extracted from different Wikipedia dumps). While we selected YAGO3 to be our background knowledge base, any multi-lingual knowledge base such as Freebase could be used as well.

- **Name-Entity Dictionary**: We compiled a dictionary from YAGO3 and Freebase to provide the potential candidate entities for each mention string. While the mention is in Arabic, the entity can belong to either the English or the Arabic Wikipedia.

- **Entity-Descriptions**: We harnessed different ingredients in YAGO3, and Wikipedia to produce a rich entity context schema. For the sake of precision, we did not employ any automated translation.

- **Entity-Entity Relatedness Model**: We fused the link structure of both the English and Arabic Wikipedia's to compute a comprehensive relatedness measure between the entities.

2 Related Work

NED is one of the classical NLP problems that is essential for many Information Retrieval tasks. Hence, it has been extensively addressed in NLP research. Most of NED approaches use Wikipedia as their knowledge repository. (Bunescu and Pasca, 2006) defined a similarity measure that compared the context of a mention to the Wikipedia categories of the entity candidate. (Cucerzan, 2007; Milne and Witten, 2008; Nguyen and Cao, 2008) extended this framework by using richer features for similarity comparison. (Milne and Witten, 2008) introduced the notion of semantic relatedness and estimated it using the the co-occurrence counts in Wikipedia. They used the Wikipedia link structure as an indication of occurrence. Below, we give a brief overview on the most recent NED systems:

The **AIDA** system is an open source system that employs contextual features extracted from Wikipedia (Hoffart et al., 2011; Yosef et al., 2011). It casts the NED problem into a graph problem with two types of nodes (mention nodes, and entity nodes). The weights on the edges between the mentions and the entities are the contextual similarity between mention's context and entity's context. The weights on the edges between the entities are the semantic relatedness among those entities. In a subsequent process, the graph is iteratively reduced to achieve a dense sub-graph where each mention is connected to exactly one entity.

The **CSAW** system uses local scores computed from 12 features extracted from the context surrounding the mention, and the candidate entities (Kulkarni et al., 2009). In addition, it computes global scores that captures relatedness among annotations. The NED is then formulated as a quadratic programming optimization problem, which negatively affects the performance. The software, however, is not available.

DBpedia Spotlight uses Wikipedia anchors, titles and redirects to search for mentions in the input text (Mendes et al., 2011). It casts the context of the mention and the entity into a vector-space model. Cosine similarity is then applied to identify the candidate with the highest similarity. Nevertheless, their model did not incorporate any semantic relatedness among entities. The software is currently available as a service.

TagMe 2 exploits the Wikipedia link structure to estimate the relatedness among entities (Ferragina and Scaiella, 2010). It uses the measure defined by (Milne and Witten, 2008) and incorporates a voting scheme to pick the right mapping. According to the authors, the system is geared for short input text with limited context. Therefore, the approach favors coherence among entities over contextual similarity. TagMe 2 is available a service.

Illinois Wikifier formulates NED as an optimization problem with an objective function designed for higher global coherence among all mentions (Ratinov et al., 2011). In contrast to AIDA and TagMe 2, it does not incorporate the link structure of Wikipedia to estimate the relatedness among entities. Instead, it uses normalized Google similarity distance (NGD) and pointwise mutual information. The software is as well available as a service.

Wikipedia Miner is a machine-learning based approach (Milne and Witten, 2008). It exploits three features in order to train the classifier. The features it employs are prior probability that a mention refers to a specific entity, properties extracted from the mention context, and finally the entity-entity relatedness. The software of Wikipedia

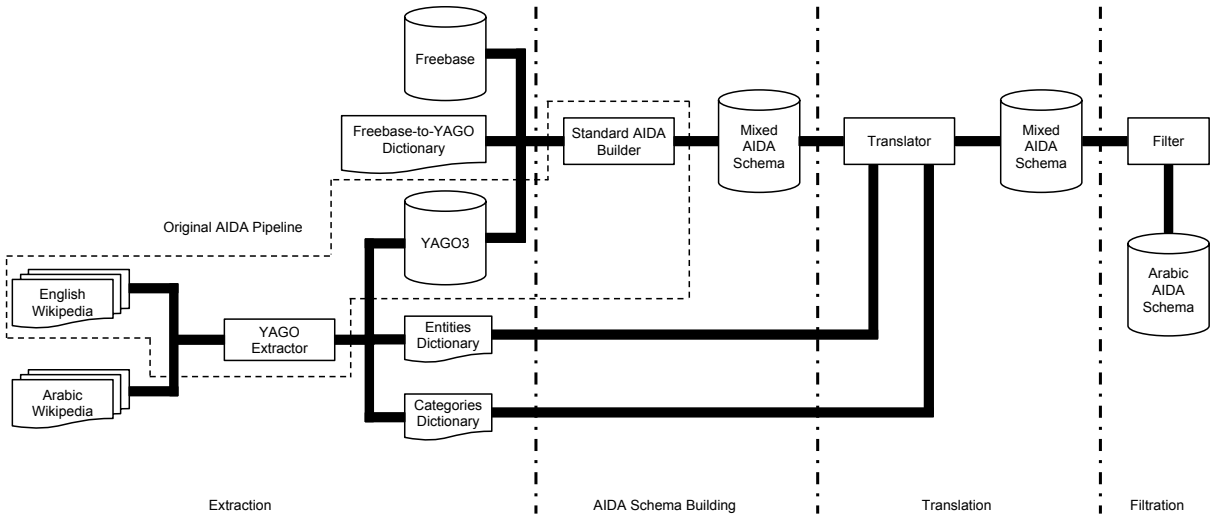

Figure 1: AIDArabic Architecture

Miner is available on their Website.

The approaches mentioned before have been developed for English language NED. As such, none of them is ready to handle Arabic input without major modification.

As of now, no previous research exploits cross-lingual resources to enable NED for Arabic text. Nevertheless, cross-lingual resources have been used to improve Arabic NER (Darwish, 2013). They used Arabic and English Wikipedia together with DBpedia in order to build a large Arabic-English dictionary for names. This augments the Arabic names with a capitalization feature, which is missing in the Arabic language.

3 Architecture

In order to build AIDArabic, we have extended the pipeline used for building an English AIDA schema from the YAGO knowledge base. The new architecture is shown in Figure 1 and indicates those components, that have been added for AIDArabic. These are pre- and post-processing stages to the original AIDA schema extractor. The new pipeline can be divided into the following stages:

Extraction

We have configured a dedicated YAGO3 extractor to provide the data necessary for AIDArabic. To this end, we feed the English and Arabic Wikipedia's into YAGO3 extractor to provide three major outputs:

- **Entity Repository**: A comprehensive set of entities that exist in both, the English and Arabic Wikipedia's. In addition, the corresponding anchortexts, categories as well as links from and/to each entity.

- **Entity Dictionary**: This is an automatically compiled mappings that captures the inter-wiki links among the English and the Arabic Wikipedia's.

- **Categories Dictionary**: This is also an automatically harvested list of mappings between the English and Arabic Wikipedia categories.

More details about data generated by each and every extractor will be given in Section 4.

AIDA Schema Building

In this stage we invoke the original AIDA schema builder without any language information. However, we additionally add the Freebase knowledge base to AIDA and map Freebase entities to YAGO3 entities. Freebase is used here solely to harness its coverage of multi-lingual names of different entities. It is worth noting that Freebase is used merely to enrich YAGO3, but the set of entities are gathered from YAGO. In other words, if there is an entity in Freebase without a YAGO counter part, it gets discarded.

Translation

Although it is generally viable to use machine translation or "off the shelf" English-Arabic dictionaries to translate the context of entities. However, we confine ourselves to the dictionaries extracted from Wikipedia that maps entities as well as categories

from English to Arabic. This is done in order to achieve a high precision derived from the manual labor inherent in interwiki links and assigned categories.

Filtration

This is a final cleaning stage. Despite translating the context of entities using the Wikipedia-based dictionaries as comprehensive as possible, a considerable amount of context information remains in English (e.g. those English categories that do not have an Arabic counterpart). To this end, any remaining leftovers in English are being discarded.

4 Implementation

This section explains the implementation of the pipeline described in Section 3. We first highlight the differences between YAGO2 and YAGO3, which justify the switch of the underlying knowledge base. Then, we present the techniques we have developed in order to build the dictionary between mentions and candidate entities. After that, we explain the context enrichment for Arabic entities by exploiting cross-lingual evidences. Finally, we briefly explain the entity-entity relatedness measure applied for disambiguation. In the following table (cf. Table 1 for details) we summarize the terminology used in the following section.

4.1 Entity Repository

YAGO3 has been specifically designed as a multilingual knowledge base. Hence, standard YAGO3 extractors take as an input a set of Wikipedia dumps from different languages, and produce a unified repository of named entities across all languages. This is done by considering inter-wiki links. If an entity in language $l \in L - \{en\}$ has an English counter part, the English one is kept instead of that in language l, otherwise, the original entity is kept. For example, in our repository, the entity used to represent Egypt is "Egypt" coming from the English Wikipedia instead of "ar/مصر" coming from the Arabic Wikpedia. However, the entity that refers to the western part of Cairo is identified as "ar/غرب القاهرة" because it has no counter-part in the English Wikipedia. Formally, the set of entities in YAGO3 are defined as follows:

$$E = E_{en} \cup E_{ar}$$

After the extraction is done, YAGO3 generates an entity dictionary for each and every language. This dictionary translates any language specific entity into the one that is used in YAGO3 (whether the original one, or the English counter part). Based on the the previous example, the following entries are created in the dictionary:

ar/مصر	→	Egypt
ar/غرب القاهرة	→	ar/غرب القاهرة

Such a dictionary is essential for all further processing we do over YAGO3 to enrich the Arabic knowledge base using the English one. It is worth noting here, that this dictionary is completely automatically harvested from the inter-wiki links in Wikipedia, and hence no automated machine translation and/or transliteration are invoked (e.g. for Person Names, Organization Names, etc.). While this may harm the coverage of our linkage, it guarantees the precision of our mapping at the same time. This is thanks to the high quality of interwiki between named-entities in Wikipedia.

4.2 Name-Entity Dictionary

The dictionary in the context of NED refers to the relation that connects strings to canonical entities. In other words, given a mention string, the dictionary provides a list of potential canonical entities this string may refer to. In our original implementation of AIDA, this dictionary was compiled from four sources extracted from Wikipedia (titles, disambiguation pages, redirects, and anchor texts). We used the same sources after adapting them to the Arabic domain, and added to them entries coming from Freebase. In the following, we briefly summarize the main ingredients used to populate our dictionary:

- **Titles**: The most natural possible name of a canonical entity is the title of its corresponding page in Wikipedia. This is different from the entity ID itself. For example, in our example for the entity "Egypt" that gets its id from the English Wikipeida, we consider the title "مصر" coming from the Arabic Wikipedia.
- **Disambiguation Pages**: These pages are called in the Arabic Wikipedia "صفحات التوضيح". They are dedicated pages to list the different possible meanings of a specific name. We harness all the links in a disambiguation page and add them as

l	A language in Wikipedia
L	Set of all languages in Wikipedia
e_{en}	An entity originated from the English WIkipedia
e_{ar}	An entity originated from the Arabic WIkipedia
e	An entity in the final collection of YAGO3
E	Set of the corresponding entities
$Cat_{en}(e)$	Set of Categories of an entity e in the English Wikipedia
$Cat_{ar}(e)$	Set of Categories of an entity e in the Arabic Wikipedia
$Inlink_{en}(e)$	Set of Incoming Links to an entity e in the English Wikipedia
$Inlink_{ar}(e)$	Set of Incoming Links to an entity e in the Arabic Wikipedia
$\underset{en \rightarrow ar}{\text{Trans}}(S)$	Translation of each element in S from English to Arabic using the appropriate dictionaries

Table 1: Terminology

potential entities for that name. To this end, we extract our content solely from the Arabic Wikipedia. For instance, the phrase "مدينة زايد" has a disambiguation page that lists all the cities that all called Zayed including the ones in Egypt, Bahrain and United Arab Emirates.

- **Redirects**: "تحويلات" denotes redirects in Arabic Wikipedia. Those are pages where you search for a name and it redirects you to the most prominent meaning of this name. This we extract from the Arabic Wikipedia as well. For example, if you search in the Arabic Wikipedia for the string "زايد", you will be automatically redirected to page of the president of the United Arabic Emirates.

- **Anchor Text**: When people create links in Wikipedia, sometimes they use different names from the title of the entity page as an anchor text. This indicates that this new name is also a possible name for that entity. Therefore, we collect all anchors in the Arabic Wikipedia and associate them with the appropriate entities. For example, in the Arabic Wikipedia page of Sheikh Zayed, there is a anchor link to the city of Al Ain "ar/العين", while the anchor text reads "المنطقه الشرقية" (in English: "The Eastern Area"). Therefore, when there is a mention called "The Eastern Area", one of the potential candidate meanings is the city of Al-Ain in United Arab Emirates.

- **Freebase**: Freebase is a comprehensive resource which comes with multi-lingual labels of different entities. In addition, there is a one-to-one mapping between (most of) Freebase entities and YAGO3 entities, because Freebase is extracted from Wikipedia as well. Therefore, we carry over the Arabic names of the entities from Freebase to our AIDA schema after mapping the entities to their corresponding ones in YAGO3.

4.3 Entity-Descriptions

The context of an entity is the cornerstone in the data required to perform NED task with high quality. Having a comprehensive and "clean" context for each entity facilitates the task of the NED algorithm by providing good clues for the correct mapping. We follow the same approach that we used in the original AIDA framework by representing an entity context as a set of characteristic keyphrases that captures the specifics of such entity. The keyphrases are further decomposed into keywords with specificity scores assigned to each of them in order to estimate the global and entity-specific prominence of this keyword. The original implementation of AIDA extracted keyphrases from 4 different sources (anchor text, inlink titles, categories, as well as citation titles and external links). Below we summarize how we adopted the extraction to accommodate the disambiguation of Arabic text.

- **Anchor Text**: Anchors in a Wikipedia page are usually good indicators of the most im-

portant aspects of that page. In the original implementation of AIDA, all anchors in a page are associated with the corresponding entity of this page, and added to the set of its keyphrases.The same holds for AIDArabic. However, we extract the anchors from the Arabic Wikipedia to get Arabic context.

- **Inlink Titles**: In the same fashion that links to other entities are good clues for the aspects of the entity, links coming from other entities are as well. In AIDA, the set of the titles of the pages that has links to an entity were considered among the keyphrases of such an entity. We pursued the same approach here, and fused incoming links to an entity from both English and Arabic Wikipedia. Once set of the incoming links was fully built, we applied - when applicable - interwiki links to get the translation of titles of the entities coming from the English Wikipedia into the Arabic language. Formally:

$$Inlink(e) = Inlink_{ar}(e) \cup \underset{en \rightarrow ar}{\text{Trans}}(Inlink_{en}(e))$$

- **Categories**: Each Wikipedia page belongs to one or more categories, which are mentioned at the bottom part of the page. We configured YAGO3 to provide the union of the categories from both, the English and Arabic Wikipedia. We exploit the interwiki links among categories to translate the English categories to Arabic. This comes with two benefits, we use the category mappings which result in fairly accurate translation in contrast to machine translation. In addition, we enrich the category system of the Arabic Wikipedia by categories from the English for entities that have corresponding English counterpart.

$$Cat(e) = Cat_{ar}(e) \cup \underset{en \rightarrow ar}{\text{Trans}}(Cat_{en}(e))$$

- **Citation Titles and External Links**: Those were two sources of entities context in the original Wikipedia. Due to the small coverage in the Arabic Wikipedia, we ignored them in AIDArabic.

Table 2 summarizes which context resource has been translated and/or enriched from the English Wikipedia.

4.4 Entity-Entity Relatedness Model

For coherent text, there should be connection between all entities mentioned in the text. In other words, a piece of text cannot cover too many aspects at the same time. Therefore, recent NED techniques exploit entity-entity relatedness to further improve the quality of mapping mentions to entities. The original implementation of AIDA used for that purpose a measure introduced by (Milne and Witten, 2008) that estimates the relatedness or coherence between two entities using the overlap in the incoming links to them in the English Wikipedia.

Despite the cultural difference, it is fairly conceivable to assume that if two entities are related in the English Wikipedia, they should also be related in the Arabic one. In addition, we enrich the link structure used in AIDA with the link structure of the Arabic Wikipedia. Hence, we estimate the relatedness between entities using overlap in incoming links in both the English and Arabic Wikipedia's together.

5 Experimentation

5.1 Setup and Results

Up to our knowledge, there is no standard Arabic data set available for a systematic evaluation of NED. In order to assess the quality of our system, we manually prepared a small benchmark collection. To this end, we gathered 10 news articles from www.aljazeera.net from the domains of sports and politics including regional as well as international news. We manually annotated the mentions in the text, and disambiguated the text by using AIDArabic. In our setup, we used the LOCAL configuration setting of AIDA together with the original weights. The data set contains a total of **103 mentions**. AIDArabic managed to annotate **34 of them correctly**, and assigned **68 to NULL**, while **one mention was mapped wrongly**.

5.2 Discussion

AIDArabic performance in terms of precision is impressive (%97.1). Performance in that regard is positively influenced by testing on a "clean" input of news articles. Nevertheless, AIDArabic loses on recall. Mentions that are mapped to NULL, either

Context Source	Arabic Wikipedia	English Wikipedia
Anchor Text	+	-
Categories	+	+
Title of Incoming Links	+	+

Table 2: Entities Context Sources

have no correct entity in the entity repository, or the entity exists but lacks the corresponding name-entity dictionary entry.

This observation confirms our initial hypothesis that lack of data is one of the main challenges for applying NED on Arabic text. Another aspect that harms recall is the nature of Arabic language. Letters get attached to the beginning and/or the end of words (e.g. connected prepositions and pronouns). In such a case, when querying the dictionary, AIDArabic is not able to retrieve the correct candidates for a mention like "بفرنسا", because of the "ب" in the beginning. Similar difficulties arise when matching the entities description. Here, many keywords do not be to match the input text because they appear in a modified version augmented with some extra letters.

6 Conclusion & Outlook

In this paper, we have introduced the AIDArabic framework, which allows named entity disambiguation of Arabic texts based on an automatically generated knowledge based derived from Wikipedia. Our proof-of-concept implementation shows that entity disambiguation for Arabic texts becomes viable, although the underlying data sources (in particular Wikipedia) still is relatively sparse. Since our approach "integrates" knowledge encapsulated in interwiki links from the English Wikipedia, we are able to boost the amount of context information available compared to a solely monolingual approach.

As a next step, intend to build up a proper dataset that we will use for a systematic evaluation of AIDArabic. In addition, we plan to apply machine translation/transliteration techniques for keyphrases and/or dictionary lookup for keywords in order to provide even more context information for each and every entity. In addition, we may employ approximate matching approaches for keyphrases to account for the existence of additional letter connected to words. As a byproduct we will be able to apply AIDArabic on less formal text (e.g. social media) which contains a considerable amount of misspellings for example. Apart from assessing and improving AIDArabic, a natural next step is to extend the framework by extractors for other languages, such as French or German. By doing so, we are going to create a framework, which will be in its final version fully language agnostic.

Acknowledgments

We would like to thank Fabian M. Suchanek and Joanna Biega for their help with adopting YAGO3 extraction code to fulfill AIDArabic requirements.

References

[Auer et al.2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th Intl Semantic Web Conference*, pages 11–15, Busan, Korea.

[Bunescu and Pasca2006] Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 9–16, Trento, Italy.

[Cucerzan2007] S. Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL 2007*, pages 708–716, Prague, Czech Republic.

[Darwish2013] Kareem Darwish. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)* , pages 1558–1567, Sofia, Bulgaria.

[Ferragina and Scaiella2010] 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 2010)*, pages 1625–1628, New York, NY, USA.

[Ferrucci2012] D. A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development (Volume 56, Issue 3)*, pages 235–249.

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

[Hoffart et al.2013] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence (Volume 194)*, pages 28–61.

[Kulkarni et al.2009] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2009)*, pages 457–466, New York, NY, USA.

[Mahdisoltani et al.2014] Farzane Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2014. A knowledge base from multilingual Wikipedias – yago3. Technical report, Telecom ParisTech. http://suchanek.name/work/publications/yago3tr.pdf.

[Mendes et al.2011] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBbpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics 2011)*, pages 1–8, New York, NY, USA.

[Milne and Witten2008] David N. Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, pages 509–518, New York, NY, USA.

[Nguyen and Cao2008] Hien T. Nguyen and Tru H. Cao. 2008. Named entity disambiguation on an ontology enriched by Wikipedia. In *Proceedings of IEEE International Conference on Research, Innovation and Vision for the Future (RIVF 2008)*, pages 247–254, Ho Chi Minh City, Vietnam.

[Ratinov et al.2011] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT 2011)*, pages 1375–1384, Stroudsburg, PA, USA.

[Yosef et al.2011] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. 2011. AIDA: An online tool for accurate disambiguation of named entities in text and tables. In *Proceedings of the 37th International Conference on Very Large Data Bases (VLDB 2011)*, pages 1450–1453, Seattle, WA, USA.