Crowdsourced Entity Markup

Lili Jiang, Yafang Wang, Johannes Hoffart, Gerhard Weikum

Max Planck Institute for Informatics
Saarbruecken, Germany
{ljiang,ywang,jhoffart,weikum}@mpi-inf.mpg.de

Abstract. Entities, such as people, places, products, etc., exist in knowledge bases and linked data, on one hand, and in web pages, news articles, and social media, on the other hand. Entity markup, like Named Entities Recognition and Disambiguation (NERD), is the essential means for adding semantic value to unstructured web contents and this way enabling the linkage between unstructured and structured data and knowledge collections. A major challenge in this endeavor lies in the dynamics of the digital contents about the world, with new entities emerging all the time. In this paper, we propose a crowdsourced framework for NERD, specifically addressing the challenge of emerging entities in social media. Our approach combines NERD techniques with the detection of entity alias names and with co-reference resolution in texts. We propose a linking-game based crowdsourcing system for this combined task, and we report on experimental insights with this approach and on lessons learned.

Keywords: Named Entity Recognition and Disambiguation, Crowdsourcing

1 Introduction

Knowledge bases, linked data, and other semantic web assets are flourishing [9, 23, 12, 26] and contribute to improved search, analytics, and recommendation services. These assets contain many billions of facts about many millions of entities like people, places, companies, music bands, songs, diseases, drugs, proteins, etc. Additional value is created by *entity-level links* that span collections, via RDF triples with the owl:sameAs predicate [9, 11]. This way, different collections complement each other. For example, while one data source knows everything about the musicians of a song, another one contains data about the sales of the song's album, and yet another one knows about the use of the song in movies or cover versions by other artists. Jointly, this allows analyzing a musician's influence on the entertainment industry.

Structured data will hardly ever be complete, as there is always some detail not captured in RDF triples and the world is rapidly evolving anyway. Therefore, it is crucial to establish also entity-level links between unstructured sources like news articles or social media and the web of linked open data. Manually creating microdata embedded in HTML pages is one approach, but this will still leave many gaps. To fill these gaps, largely automated methods are needed, discovering names of entities in text, tables, or lists of surface web contents and mapping them to entities in linked-data collections. As names can have many different meanings, this entails the need for *Named Entity Recognition and Disambiguation (NERD).*

Fully automatic NERD is inherently difficult and may also be computationally expensive (see, e.g., [18, 15, 10, 22, 13, 2]). NERD performs very well for prominent entities in high-quality texts like news articles, but they degrade in precision and recall when dealing with long-tail entities and difficult inputs like social media. Since advanced methods utilize machine learning or extensive statistics for semantic relatedness measures among entities, the availability of labeled training data is usually a big bottleneck. This is one of issues where crowdsourcing [4] can help, in order to improve NERD quality.

Even if we had perfect NERD methods, the cross-linkage between unstructured web contents and semantic data collections would still have big gaps. The reason is the *dynamics* of the world: new entities come into existence (e.g., songs, hurricanes, scandals) and unnoted entities suddenly gain importance (e.g., Edward Snowden, Adele two years ago). When facing such *emerging entities*, we cannot map them to a knowledge base (yet) as there are no RDF triples about them. However, we can capture their mentions under different names and try to gather equivalence classes of text phrases that refer to the same entity. This is known as the task of *coreference resolution (CR)* (see, e.g., [8, 20, 21, 24]). For example, we should discover the mentions "Edward Snowden", "NSA agent Snowden", and "the Prism whistleblower" and infer that they denote the same emerging entity, while also inferring that "actress Snowden" and "CEO Snowden" are separate entities.

CR methods can also help to increase the recall of NERD for known entities, by capturing more surface phrases (e.g., [17, 19]). For example, the German football team FC Bayern Munich may be known and detectable as "Bayern Munich", "FC Bayern', or as "Germany's most successful football club", but the additional name "triple winner" makes sense only since end of May 2013 (when the team won three major championships). If, for a given text, we infer that "triple winner" and "UEFA champion 2013" are the same entity, we can map more text mentions onto entities, thus improving NERD recall at high precision. Systematically gathering alias names for entities is the problem of *alias detection (AD)*. It has been studied in the literature, harnessing href anchor texts, click logs, and other assets (see, e.g., [14, 25]). However, doing this for emerging entities that are not yet registered in a knowledge base is a largely unexplored task.

The goal of this paper is to address the above problems in creating semantic markup for entities. Our approach is unique in that we address the three problems NERD, CR, and AD in a joint manner. Our methodology is *crowdsourcing*: asking people to annotate text snippet (e.g., tweet). While this approach may seem straightforward, it does come with technical challenges. First, we need to cast the problem into simple user interactions so that laymen can contribute with little effort. Second, we need to cope with highly varying quality of user contributions. Third, we need to optimize the benefit/cost ratio, by making judicious choices about which text snippets are shown to which people.

This paper presents a first cut on these problems, including experimental studies. The benefit of our crowdsourcing architecture is twofold: i) we create semantic markup in the form of co-reference between mentions, which can be directly used as input for methods that connect the web of unstructured contents with the web of linked data at the entity level, and ii) we lay the foundation to use this annotated contents to improve automated methods for NERD, CR, and AD. In the future, by continuously running a low-cost crowdsourcing process on news or social media, we can periodically re-train and re-configure automated methods and adjust them to the dynamics of web contents.

2 Related Work

NERD methods [18, 15, 10, 22, 13, 2] aim to identify entity mentions in natural-language text and weakly structured web contents like HTML tables and lists, and link the mentions to entities registered in a knowledge base or linked data source. Coreference resolution (CR) identifies mentions in text that refer to the same entity [8, 20, 21, 24], but without mapping them onto data or knowledge bases. Note that these tasks are fairly different from database-oriented task of entity resolution, aka. entity matching or record linkage [7], which is solely focused on structured records (with known schema) as input.

Crowdsourcing [4, 16, 5, 3] harnesses human input for tasks that are inherently difficult for computers, such as image tagging or language understanding. Approaches along these lines come in two major families: i) explicit crowdsourcing with HITs (human intelligence tasks) assigned to paid workers on platforms like Amazon Mechanical Turk (`www.mturk.com`) or CrowdFlower (`crowdflower.com`), and ii) implicit crowdsourcing where the task is piggybacked on human-computer interactions or in the form of a game.

Crowdsourcing was used for the problem of entity resolution [27] on structured database records. Recall that this task is quite different from our problem of NERD and CR over text snippets. This work also compares a list-wise with a pair-wise style user interface. In contrast, we aim to compare user behavior under different user interfaces (i.e., pair-wise and linking-game based interface).

3 Overview of Methodology

We have developed a framework for combining NERD, CR, and AD. Figure 1 gives a pictorial overview. The emphasis in this paper is on crowdsourcing the task of CR, in the form of a linking game, and harness the user feedback obtain this way for improving AD and NERD.

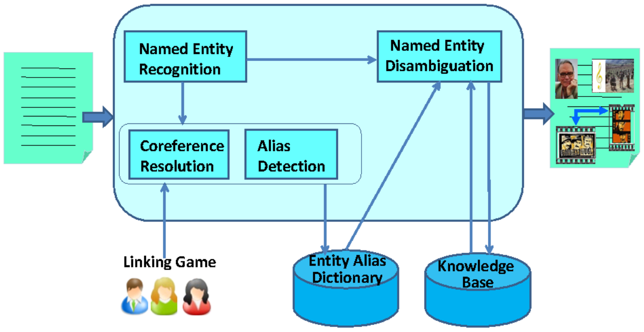


Fig. 1. Framework for NERD, CR, and AD.

In the following we briefly characterize the functionality of each component, and explain the dataflow between components.

Named Entity Recognition (NER). The input text is processed to discover *mentions* of named entities, that is, surface phrases that are likely to denote individual entities (as opposed to common noun phrases). Our implementation currently uses the Stanford NER Tagger [6] for this purpose (a trained CRF).

Crowdsourced Coreference Resolution (CR). All mentions in the same input text are highlighted and presented to human players, using a game-like interface. The participating users are asked to connect mentions that refer to the same entity. This way we obtain equivalence classes of mentions. Note that this does not perform any disambiguation yet: we still do not know which entity an equivalence class of mentions refers to, and in the case of newly emerging entities may not have the proper entity registered in our knowledge base anyway.

Alias Detection (AD). The CR step has the benefit of providing us with alias names for the same entity. Some of these names may already be present in our dictionary of entity aliases (e.g., “the US president’s wife” for Michelle Obama), but others are new discoveries (e.g., “the First Lady of the White House”). If we can later, in the NED step, map the entire equivalence class of coreferences to an entity, we can easily add the new aliases to the dictionary. This way, we improve the AD task and increase the coverage of our dictionary.

Named Entity Disambiguation (NED). Finally, we attempt to map all mentions to canonicalized entities registered in a knowledge base. We use the YAGO knowledge base for this purpose (`http://yago-knowledge.org`), but can easily switch to other choices like DBpedia or Freebase. The actual NED computation is based on the AIDA method [10] and its open-source software (`https://github.com/yago-naga/aida`). AIDA combines context-similarity measures with coherence measures for the entities chosen for different mentions. We have further extended AIDA to become aware of the coreference equivalence classes obtained in the CR step. This extension is presented in Section 5.

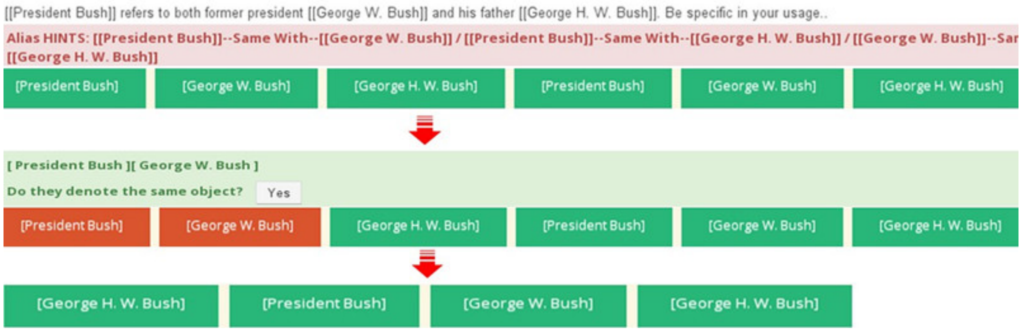
4 Crowdsourced Coreference Resolution

4.1 Mention Linking Game

We created a crowdsourcing interface that allows humans to highlight coreferenced mentions in a text snippet in a light-weight manner. To minimize the burden on humans and as an additional incentive, we developed a game-like interface inspired by the “Linking Game”[1], in which players earn points by finding identical icons in an image. This in turn is reminiscent of the well-known Concentration Game, also known as Memory, just with all cards already open.

Figure 2 shows a sequence of three screenshots of our mention linking game. Players are asked to mark up all co-referent mentions for a given set of mentions highlighted in the text. The user receives hints about which mentions may possibly be equivalent, using simple heuristics for automated CR. All mentions are then presented as green blocks for markup by the user. When the user selects blocks, they are turned red. Once the user clicks on “Yes” to confirm that they are coreferences, these blocks are removed from the

[1] `http://www.appszoom.com/android_games/sports_games/cute-puppys-link-game_bsddz.html`



[[President Bush]] refers to both former president [[George W. Bush]] and his father [[George H. W. Bush]]. Be specific in your usage..

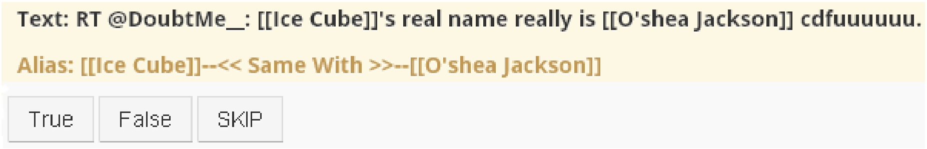
Alias HINTS: [[President Bush]]--Same With--[[George W. Bush]] / [[President Bush]]--Same With--[[George H. W. Bush]] / [[George W. Bush]]--Sar
[[George H. W. Bush]]

[President Bush][George W. Bush]

Do they denote the same object?

Fig. 2. Linking-Game Interface

user's view. When players are very certain about one selection, they can select the same equivalent mention pair multiple times. This gives us an implicit way of estimating the confidence of a user's input.



Text: RT @DoubtMe_: [[Ice Cube]]'s real name really is [[O'shea Jackson]] cdfuuuuuu.

Alias: [[Ice Cube]]--<< Same With >>--[[O'shea Jackson]]

Fig. 3. Pair-wise User Interface

To compare the effectiveness of the linking-game based interface against more traditional crowdsourcing interfaces, we also designed game UI for judging each pair of mentions separately, as shown in Figure 3. A pair of mentions is presented, and the player has to make one of the three choices: Yes, No, or Skip.

4.2 Quality Control

For assessing the quality of the players, we prepared a set of gold-standard texts for which we identified the correct equivalence classes of mentions. These gold-standard texts are occasionally presented as linking-game tasks, and a user's performance on these is a first-cut estimate for the confidence in the user's markup.

4.3 Feedback for Automated Coreference Resolution

High-confidence annotations obtained from the game are chosen as the crowdsourced results of CR. These results are directly used to enhance named entity disambiguation, as described in the following Section. Additionally, high quality annotations can be used as training data. The samples will help to better learn feature weights, where features could be alias matching, abbreviation/acronym matching, string similarity, position relative to the two mentions of interest, part-of-speech tags, etc. Details of this enhancement and its performance are beyond the scope of this paper.

5 Combining NERD, CR, and AD

We used the AIDA tool [10] as a basis for our crowdsourcing-enhanced NERD method. AIDA works in four steps. First, it uses the Stanford NER Tagger to identify mentions in the input text. Second, it generates candidate entities by looking up the surface names in the dictionary and retrieving the associated entities from the knowledge base. Third, it builds a graph that connects mention nodes with candidate entity nodes by edges that are weighted with context-similarity scores, and connects pairs of candidate entity nodes by edges that are weighted with semantic coherence scores. Fourth and last, AIDA runs an algorithm for computing a dense subgraph whose entity nodes yield the desired disambiguation. Figure 4, upper part, shows an example graph with these two types of edges. The graph contains a third kind of edges, connecting pairs of mention nodes. These are actually added by our crowdsourced-CR process, as explained in Section 5.

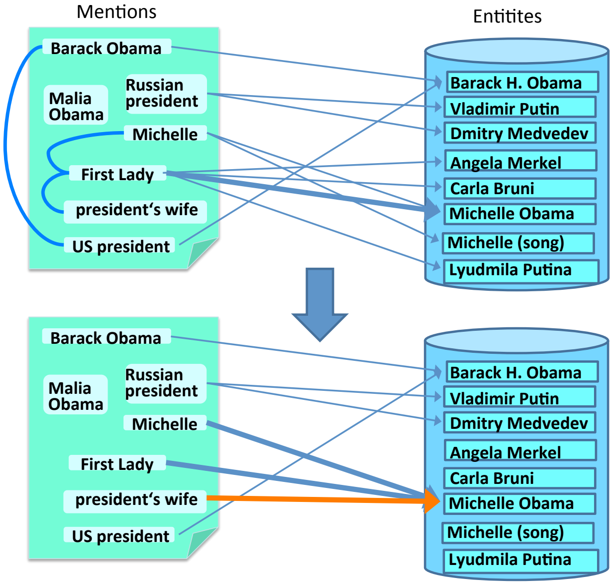


Fig. 4. Example graph for combined NED and CR

In the example, “Michelle” is a highly ambiguous mention, which is difficult to map to the proper entity. Here, the crowdsourced CR yields valuable input by linking this mention with the other two mentions “president’s wife” and “First Lady”, thus easing the tasks of NED. Note that some of the mentions marked up in the NER step may not be in the dictionary; so usually no candidate entities would be generated for a mention such as “president’s wife”. By the CR markup from the crowdsourcing phase, we can transfer the candidate entities from other mentions, “First Lady” and “Michelle”, to this newly recognized phrase. We actually choose the entity that has the highest weight among all the candidates in the same CR equivalence class for all the mentions. Finally note that one mention in the example text, “Malia Obama” is not linkable to the knowledge base at all, as there there is no suitable entity there.

Our enhancements of AIDA work by extending the mapping graph. For every set of mentions, $m_1, m_2, \ldots, m_k$, that were combined into one equivalence class by the crowdsourced CR, we proceed as follows:

- Case 1: All of $m_1, m_2, \ldots, m_k$ have matches in the dictionary. In this case, we generate all respective candidate entities, by lookups in the knowledge base, and then choose the highest weighted entity among all candidate sets, retaining only this entity for all mentions in the CR equivalence class.
- Case 2: The set $M = \{m_1, m_2, \ldots, m_k\}$ contains some mentions that do not have any matches in the alias dictionary, say subset $N \subset M$. In this case, we determine the entity for the potentially linkable mentions, subset $L = M - N$, according to Case 1 and then add it to all mentions in N.
  In addition, we insert the mentions in N as new alias names for the retained candidate entities into the alias dictionary, thus enhancing the AD component of our framework.
- Case 3: None of the mentions in $M = \{m_1, m_2, \ldots, m_k\}$ has any match in the dictionary, so they are all non-linkable. In this case, we drop these mentions from the NED graph. However, we do insert this set of mentions into the alias dictionary as alias names for an unknown entity. This can pay off later, for a new input text, if that text has a CR equivalence class that includes both a name associated with a known entity and an alias from M. This way, we potentially improved both AD and NERD in the long run.

The lower part of Figure 4 shows the graph that results from these steps. After these graph-enhancement steps, all mention-mention edges are removed. The resulting graph can be directly fed into the AIDA tool for the actual NED computation.

6 Experimental Results

6.1 Experimental Setup

In our preliminary studies reported here, we focus on two types of entities from tweets: *persons* and *locations*. We used lists of 50 US states and 50 celebrities, from the prior work of [1] (`http://www.iba.t.u-tokyo.ac.jp/~danushka/data/aliasdata.zip`). Each entity comes with a small number of alias names. For example, Michael Jordan (the basketball player) has alias names "Air Jordan", "His Airness", and "MJ", and Whoopi Goldberg is also known as "Da Whoop" and "Caryn Elaine Johnson".

We further extended this dataset in three ways. First, we included additional persons (all US presidents) and locations (a set of large cities around the world) as concerned entities. This led to a total of 93 person entities and 150 location entities. Second, we gathered tweets from Twitter (`twitter.com`) by generating queries with the entity names and their alias names. Third, we added tweets from the UK election 2009. We selected 140 tweets for crowdsourcing experiment, and 100 tweets for NED evaluation. The number of mentions are counted by using a liberal NER method, combining the Stanford Tagger [6] and a dictionary-based matcher for entity names and aliases. Our complete experimental data is available at `http://www.mpi-inf.mpg.de/yago-naga/aida/download/iswc-crowdsem2013.zip`.

6.2 CR Performance

A total of 14 university students participated in our crowdsourcing experiment, 7 playing the linking game and 7 using the pair-wise UI. For evaluation, we manually annotated 140 tweets. We aggregated the human contributions for the same tweet by weighted voting, where weights reflect the confidence in a user (which in turn is based on how well the user performed for the occasional gold-standard inputs, see Section 4.2). We compared the two crowdsourcing settings against a fully automated heuristic algorithm for CR, based on the following simple rules:

- When two mentions exactly match aliases for the same entity in our dictionary, the algorithm connects them into a CR equivalence class.
- When two mentions have high string similarity above a threshold, the algorithm connects them.
- When the text between two mentions contains a strong pattern such as “also known as”, “called”, “referred to”, etc., the algorithm connects them.

The results in terms of precision, recall, and F1 scores are shown in Table 1. We observe that the Linking-Game-based crowdsourcing clearly outperformed the pair-wise annotator UI. This is due to the vastly increased number of decisions necessary for pair-wise annotators, which increases the risk of making mistakes. The game-based crowdsourced CR also won against the rule-based algorithm by a large margin, in terms of F1 scores. However, the experiment also revealed trade-offs: the automatic algorithm did much better in terms of recall, but was much inferior to the crowdsourced CR in terms of precision.

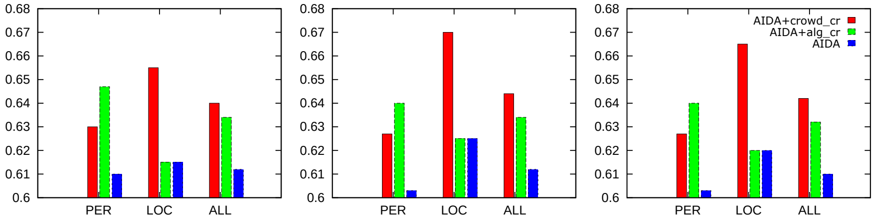
Mention Type	**Linking Game**			**Pair-wise UI**			**Algorithm**		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
person	**0.85**	0.70	0.77	0.52	0.80	0.63	0.53	**1.0**	0.69
location	**0.98**	0.81	0.88	0.61	0.54	0.57	0.58	**1.0**	0.73
overall	**0.92**	0.76	0.83	0.56	0.67	0.60	0.55	**0.99**	0.71

Table 1. Linking-Game vs. Pair-wise-UI vs. Algorithm Results for CR

6.3 NED Performance

We manually mapped the mentions in 100 tweets onto proper entities for as ground-truth for experiments on NED performance. We compared three methods: the standard AIDA method, our enhancement using crowdsourced CR annotations (see Section 5), an analogous enhancement of AIDA by CR annotations obtain from the rule-based heuristic algorithm (see CR experiments above). The results are shown in Figure 5.

The results clearly show that the combined CR+NED approach (AIDA+alg_cr) achieves much better performance than the state-of-the-art NED method (AIDA) alone. When comparing the influence of crowdsourced CR vs. algorithmic CR, we see mixed results: none of these two methods dominates the other. However, in terms of overall F1 score across all mentions, the crowdsourcing-enhanced method (AIDA+crowd_cr) is the overall winner.



(a) Precision (b) Recall (c) F1 Measure

Fig. 5. NED Performance Comparison

7 Lessons Learned

This paper presented a new approach to combining NED (Named Entity Disambiguation), CR (Coreference Resolution), and AD (Alias Detection) with crowdsourcing-based CR. Our experiments are a first proof of concept that this directions is worthwhile being pursued further at larger scale. The Linking-Game-based interface turned out to yield better results than a more traditional annotator UI. This is an encouragement towards intensifying and extending this game-based approach.

As for the overall improvement that CR contributes to NED performance, our experiments, albeit still small-scaled, clearly indicate that CR annotations are very beneficial for NED. Moreover, they also contribute to maintaining the alias name dictionary and thus handling emerging entities. As for the crowdsourced vs. algorithmic CR (see Table 1), the situation is less clear, though. The crowdsourcing approach has both higher precision and recall, however, it still has weaknesses when text snippets are very demanding. For example, consider the tweet: “The Rich are Running from California. The once Golden State is trying to bail itself out by going after the rich.” Realizing that “California” and “Golden State” denote the same entity was beyond what our crowdsourcing users could do, so our approach failed on this sample. Co-occurrence statistics for mentions, mined from Web and text corpora, could overcome this weakness. This calls for a new hybrid between crowdsourced and algorithmic methods.

8 Acknowledgements

This work is supported by the 7th Framework IST programme of the European Union through the focused research project(STREP) on Longitudinal Analytics of Web Archive data (LAWA) under contract no. 258105.

References

1. D. Bollegala, T. Honma, Y. Matsuo, and M. Ishizuka. Automatically extracting personal name aliases from the web. In *Proceedings of the 6th international conference on Advances in Natural Language Processing*, GoTAL ’08, pages 77–88, 2008.
2. M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *WWW*, pages 249–260, 2013.
3. G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.

4. A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, 2011.
5. T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. In *CSLDAMT*, pages 80–88, 2010.
6. J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, ACL '05, pages 363–370, 2005.
7. L. Getoor and A. Machanavajjhala. Entity resolution: theory, practice & open challenges. *Proc. VLDB Endow.*, 5(12):2018–2019, Aug. 2012.
8. A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *EMNLP*, pages 1152–1161, 2009.
9. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
10. J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, EMNLP '11, pages 782–792, 2011.
11. A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of linked data conformance. *J. Web Sem.*, 14:14–44, 2012.
12. E. H. Hovy, R. Navigli, and S. P. Ponzetto. *Special Issue on Artificial Intelligence, Wikipedia and Semi-Structured Resources*, volume 194. 2013.
13. R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11), 2012.
14. L. Jiang, J. Wang, P. Luo, N. An, and M. Wang. Towards alias detection without string similarity: an active learning based approach. In *SIGIR*, pages 1155–1156, 2012.
15. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–466, 2009.
16. E. Law and L. von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.
17. T. Lin, Mausam, and O. Etzioni. No noun phrase left behind: Detecting and typing unlinkable entities. In *EMNLP-CoNLL*, pages 893–903, 2012.
18. D. N. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518, 2008.
19. N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *ACL*, page to appear, 2013.
20. A. Rahman and V. Ng. Coreference resolution with world knowledge. In *ACL*, pages 814–824, 2011.
21. L.-A. Ratinov and D. Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *EMNLP-CoNLL*, pages 1234–1244, 2012.
22. L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, pages 1375–1384, 2011.
23. N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, W. Hall, and m. c. schraefel. Linked open government data: Lessons from data.gov.uk. *IEEE Intelligent Systems*, 27(3):16–24, 2012.
24. S. Singh, A. Subramanya, F. C. N. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL*, pages 793–803, 2011.
25. V. I. Spitkovsky and A. X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.
26. F. M. Suchanek and G. Weikum. Knowledge harvesting in the big-data era. In *SIGMOD Conference*, pages 933–938, 2013.
27. J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endowment*, 5(11):1483–1494, July 2012.