Entity Linking: Test Collections Revisited

Laurent Mertens, Thomas Demeester, Johannes Deleu, Matthias Feys, Chris Develder
Ghent University - iMinds, Belgium

firstname.lastname@intec.ugent.be

ABSTRACT

This paper analyzes two important conditions that are usually taken for granted in the evaluation of information retrieval systems: the test queries should be representative for the intended application scenario, and a sufficient amount of queries are needed to robustly assess system performance, as well as discern performance differences between systems. Both issues have important consequences, as studied in this paper for the specific case of Entity Linking systems. We investigate two methods for automatic query generation, and show them to have a vast impact on evaluated system performance. We further demonstrate the effect a query set's size has on its ability to faithfully distinguish systems, and propose a method for assessing the possible impact on system performance that adding a specific number of queries to the set might have.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Miscellaneous

Keywords

Evaluation, Entity Linking, query selection

1. INTRODUCTION AND BACKGROUND

Evaluating system performance on Information Retrieval or Extraction tasks has always been a difficult yet important subject. Typically, this involves the creation of an annotated test set, the *Golden Rule* (GR), and the definition and application of appropriate evaluation measures. With increasing sizes of test collections came the problem of incomplete assessments, and its effect on retrieval measures. This problem has been explored, e.g., in [6]. Another issue is the inter-assessor disagreement in test collections, studied in, e.g., [2] and [4].

In this paper, we study the problem of choosing the number and type of test queries for the task of Entity Linking (EL), and its effect on system performance evaluation. EL involves mapping named entities in textual documents, whose surface forms in a particular text we refer to as *mentions*, to their corresponding entry in an external Knowledge Base (KB) if such entry exists, or indicate its absence otherwise, using the keyword "NIL", Not In List.

Prior to the advent of the Text Analysis Conference[1] (TAC), the few works that appeared on EL exploited the specific Wikipedia link structure to automatically generate annotated queries, or used their own, often small, datasets. The former approach was used, e.g., in the seminal works [1] and [3]. A prime example of the latter approach is again [3], that besides Wikipedia used a set of 100 news articles as development data, and 20 news articles for post-hoc evaluation of the 756 surface forms that were correctly identified by the used Named Entity Recognition (NER) system.

Since the creation of the TAC Entity Linking task in 2009, its query sets have become a standard benchmark. These query sets contain manually selected and annotated mentions, mainly from news articles, but also from, e.g., blogs and discussion forums. Since 2011, next to the standard English EL task, a Cross-Lingual EL task has been organized, to link a mention in a non-English (i.e., Chinese or Spanish, resp. since 2011 and 2012) document to an entry in the provided English KB. Also, since 2011, simple NIL detection has been extended to "*cluster queries referring to the same non-KB (NIL) entities and provide a unique ID for each cluster*"[2]. Since 2012, TAC queries have been focusing extensively on very difficult cases.

Although the various aforementioned initiatives created valuable test collections, the impact of their size and properties (e.g., number of in-KB queries, or links, vs. NILs) on system performance has never been specifically addressed in literature. This paper fills this gap, with the following contributions:

- We show how different query selection approaches heavily impact system evaluations, and should be related to the intended application scenarios. (Section 2)
- We quantify the influence of the query set size on EL system performance. (Section 3)

2. QUERY SELECTION STRATEGIES

We define an EL test collection as a set of queries, where each query consists of at least the following elements:

- A document identifier.
- A mention that belongs to the document.
- The entity to which this mention should be resolved, or the NIL identifier.

In the context of EL, this means that a system should be presented the specified mention-document pair to be resolved, and that afterwards, the system output should be compared to the given Golden

[1] http://www.nist.gov/tac/

[2] As per the TAC 2011 Mono-Lingual Entity Linking task description.

FIRE '14 Bangalore, India

Rule for correctness. We adhere to the "one meaning per document" assumption, stating that multiple occurrences of a particular mention in a specific document refer to a single entity, and do not focus on NIL-clustering.

In a large dataset, one can distinguish two clear extremes as far as entity presence is concerned. On the one hand, the dataset will contain popular mentions that occur very frequently and are (mostly) relatively easy to resolve to the corresponding entity (e.g., popular politicians in news archives). On the other hand, there is a large amount of mentions that only appear very few times over the entire corpus, and whose underlying entity is often not present in the KB. We propose two ways of automatically selecting mentions from a given dataset, and relate these methods to the discussed extreme cases. Note that these methods deal with the selection of query *mentions*. In all cases, Golden Rule annotations should of course be made by human assessors.

Denote the entire set of documents constituting the corpus as D, and the complete set of mentions present in the corpus as M. Further define M_d as the set of all mentions $m \in M$ occurring in a particular document $d \in D$, and, inversely, D_m as the set of documents that contain a particular mention $m \in M$. Using these definitions, we define the following random query mention selection methods, to construct a test collection:

Random Document (RD).

A first way is to first randomly select a document $d \in D$ according to a uniform distribution, followed by a uniform random selection of a mention $m \in M_d$. We expect this method to harvest queries that primarily focus on mentions that are strongly present in the corpus, and hence not only have a much higher probability of reflecting a known entity, but also that an EL system will have less trouble correctly resolving these mentions.

Random Mention (RM).

A second way is to first use random uniform selection to pick a mention $m \in M$, followed by a uniform random selection of a document $d \in D_m$. We expect this method to harvest queries mostly from the "long tail" of mentions, i.e., mentions that rarely occur in the corpus, since most mentions only appear in very few documents. Hence, these mentions are less likely to reflect a known entity, and even if they do, reflect more obscure entities that are harder to correctly resolve to.

2.1 Evaluation: System Description

We carried out our schemes on a one-year news archive of Dutch news articles from 2011, containing around 750,000 documents. The number of queries gathered per method and per type (link or NIL) is listed in Table 1. Next to the RD and RM mentions, the full query set also contains a number of manually selected queries, where a balance between intuitively hard and easy cases was pursued, with the focus on links (as opposed to NILs). The entity linking system used for evaluation is a port for the Dutch language of the system we originally developed for participation in the TAC 2013 English EL task[3]. Additionally, the English NER systems originally used were replaced with a Dutch in-house NER system. Our TAC system is described in detail in [5]. It is a rule-based system, which follows a standard candidate selection → candidate scoring → NIL detection overall scheme. Individual rules, e.g., for candidate selection, are optimized on a per label basis for locations, organizations and persons. The features used for scoring are mainly overlap between different facets of Wikipedia and the considered article. These features are then combined using a vector weight model. The highest scoring candidate, if present, is finally checked against a NIL detection scheme involving several thresholds (e.g., simple score threshold, ratio of highest score to second-highest score, etc.), which allows transforming its score into a binary NIL detection score.

[3]This system can be made available to researchers. Please contact the authors for further information.

	Manual	RD	RM	Total
Link	367 (73%)	753 (54%)	169 (18%)	1289 (45%)
NIL	133 (27%)	654 (46%)	777 (82%)	1564 (55%)
Total	500	1407	946	2853

Table 1: Number of queries per selection strategy: manual, Random Document (RD), and Random Mention (RM).

	Manual		RD		RM	
	P	R	P	R	P	R
Link	91.1%	86.6%	89.1%	80.5%	81.9%	56.2%
NIL	74.2%	84.2%	83.2%	92.5%	91.6%	97.8%
Total	87.1%	86.0%	87.0%	86.1%	89.8%	90.4%

Table 2: Precision (P_L, P_N, and P) and recall (R_L, R_N, and R) on RD and RM query subsets.

2.2 Evaluation: Results

We define the following *recall* and *precision* metrics:

- Link Recall (R_L): percentage of correctly resolved link queries.
- NIL Recall (R_N): percentage of correctly resolved NIL queries.
- Total Recall (R): weighted average of $R_L$and R_N, with as weights the number of corresponding queries in the GR.
- Link Precision (P_L): amount of correctly predicted links over number of predicted links by the system.
- NIL Precision (P_N): amount of correctly predicted NILs over number of predicted NILs by the system.
- Total Precision (P): weighted average of $P_L$and P_N, with as weights the number of corresponding queries in the GR.

The F1 measure is the usual harmonic mean of R and P.

Table 1 shows that the number of links in the RM subset is significantly lower than for the RD subset, only constituting 17.9% of all RM queries, compared to 53.5% for RD queries. This confirms that when randomly choosing a mention out of M, the probability of selecting a mention whose underlying entity is represented in the KB is significantly lower compared to when one first randomly chooses a document from D.

System performance on RD and RM queries is shown in Table 2. The much lower recall for RM links compared to RD links, indicates that these links are significantly more difficult to resolve correctly, providing positive evidence for our assumption that these entities are typically less popular.

This suggests that if one is interested in building an EL system that will satisfy a casual user's need, focus on the RD queries, reflecting more popular entities, will probably be advisable. Instead, if one wants to build a system able to find the proverbial needle in the haystack, focus on the RM queries will prove beneficial.

3. INFLUENCE OF QUERY SET SIZE

We now turn to the question of assessing the influence of the size of the used query set on performance metrics (P, R, F1). To motivate this question, we used submitted runs for the TAC 2013 EL task. Figure 1 shows the F1 evolution for increasing query set size for all 27 participating teams' best scoring run, whereby the (arbitrary) query ordering as provided by TAC was maintained. This graph clearly shows that below a certain threshold in query set size,

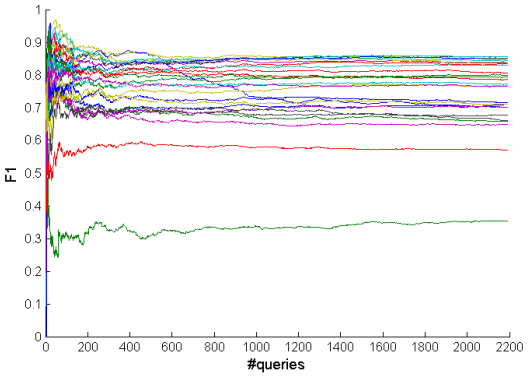

Figure 1: Evolution of F1 as function of query set size for top scoring TAC 2013 EL system runs.

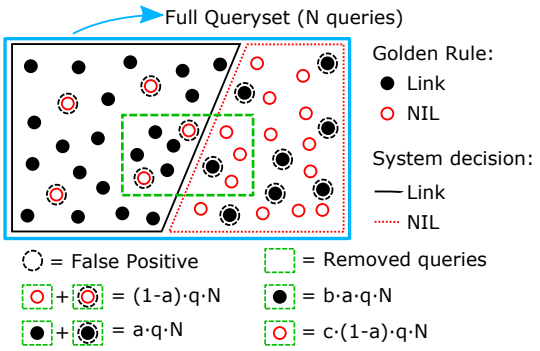

Figure 2: Illustration of the a, b and c parameters: amount of test queries according to Golden Rule and system decision.

system performance fluctuates greatly. The right half of the graph seems to show stabilized system performance, and as a result, also a more or less fixed ranking in systems (according to F1 performance). But is this really the case?

Concretely, we wish to answer two questions, namely (1) what is the possible impact on a single system's performance of adding or removing a number of queries to a GR, and (2) how faithfully can a given GR be used to distinguish between different systems? Since annotating queries is a labor intensive task, ideally we wish to assess the effect that adding a certain number of queries might have on system performance measures. The assumption we make is that the used query set is a faithful representation of the total universe of queries, and that for a "sufficiently small" amount of removed queries, the effect will be comparable to that of adding the same amount of queries. Thus, we investigate the influence on P, R, and F1 metrics of a system upon random removal of a certain fraction q of queries.

To model this problem, two elements need to be known: the GR properties, and system performance over this GR according to the metrics defined in Section 2.2. By GR properties, we understand the amount of queries it contains, as well as the fraction of links and NILs. The question we now ask ourselves is the following: when randomly removing a query from the GR, what are the chances that it exhibits a particular set of properties? In our case, the concrete possibilities are: a) the query is either a GR NIL or link, and b) the query is either correctly or wrongly resolved by the system. There are two important caveats to what seems an otherwise straightforward problem. First, whenever we remove a query from the set, we alter the properties of the remaining set, especially if the query set size is limited. As a consequence, the probabilities of picking either a link or NIL do not follow a binomial distribution, but a hypergeometric distribution, and similarly so for picking a query that is correctly or wrongly resolved. Second, links behave in a more complex way than NILs. Indeed, whenever a system resolves a GR NIL query to NIL, it is automatically correct, but when a system resolved a GR link to an entity, we have no guarantee that it resolved it to the *correct* entity. This uncertainty makes for a possible spread in effect on the resulting system performance. Consider, e.g., that a number of GR links are removed that are also resolved as links by the system, than performance will be affected best/worst if all links were wrongly/correctly resolved respectively.

To compute the minimum and maximum effect of removing a fraction q of a total of N queries (i.e., "best" and "worst" cases), as well as simulating the effect of randomly picking those queries (i.e., picking a random position between these two extremes), one needs to specify three properties of the removed queries: (i) the fraction a of removed queries that are GR links, (ii) the fraction b of those removed links that are also resolved as links by the system, and (iii) the fraction c of the $(1-a)\cdot q\cdot N$ removed NIL queries that the system correctly resolves as NIL[4]. This directly allows to derive the modified test collection's properties, and the various evaluation measures[5]. For worst case, take all b queries to be resolved correctly, and inversely for best case. When one is interested in generating a random system behavior rather than min and max limits, one needs to randomly generate which fraction of the b queries were resolved correctly, a quantity which also follows a hypergeometric distribution. Note that a, b and c are subject to constraints, e.g., we cannot remove more NIL true positives than the system predicts. Figure 2 illustrates the relation of these parameters to the query set.

Note that this model is applicable to all types of problems that exhibit the same characteristics as the EL problem, i.e., can be cast as two consecutive classification problems. Indeed, abstractly speaking, the EL problem can be cast as a binary classification, possibly followed by a multiclass classification. For EL this becomes for a given mention m (i) determining whether the entity m is known in the KB, and if so (ii) resolve m to the correct KB entry. Further note that the hypergeometric distributions exactly describe the system behavior when removing a subset of queries. However, if the change in proportions of the different parameters is neglected, binomial distributions can be used, also for the prediction of the behavior when new queries are added, rather than removed.

3.1 Experimental Verification

In order to assess the validity of our theoretical model, we carried out the following experiment. For each query subset of the query set described in Section 2 (Manual, RD and RM), we randomly removed a fraction q of all queries and computed the resulting F1 on the reduced query set a total of 10,000 times. Similarly, we used our theoretical model, tuned to simulate a system with our system's performance statistics on the GR, to compute 10,000 predicted F1 values after removal of q queries. For each iteration, we randomly generated all parameters. A comparison of the distributions of the resulting F1 values on each query subset for $q \in \{5\%, 20\%\}$ is depicted in Fig. 3, and the correspondence between theory and experiment confirms the validity of our model.

3.2 Influence of Query Set Size

[4]This is the system's recall for those queries.

[5]Author's derivation on request.

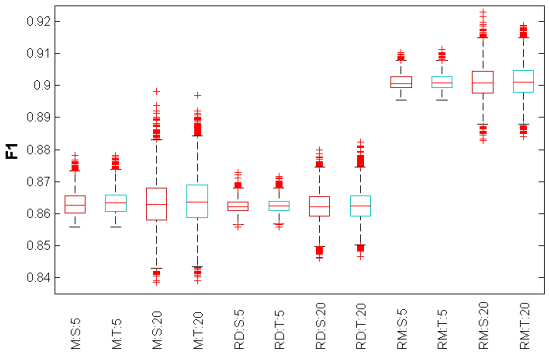

Figure 3: F1 distribution comparison between system (S) and theory (T). M = manual, RD = Random Document, RM = Random Mention, x = removal of x% of queries.

		Mean	Std	Min	Max
$X = 10$	Set1	0.876	0.000	0.875	0.877
	Set2	0.876	0.001	0.874	0.882
$X = 50$	Set1	0.876	0.001	0.873	0.879
	Set2	0.876	0.002	0.869	0.888
$X = 100$	Set1	0.876	0.001	0.872	0.881
	Set2	0.876	0.003	0.865	0.893

Table 3: F1 distribution after removal of X queries for query set size 2853 (Set1) and 1000 (Set2). Std = Standard deviation, Min/Max = observed min/max F1 values.

To estimate the effect of the size of the used GR on system performance, we essentially perform the same experiment as described in Section 3.1 for the theoretical model, only this time removing a discrete number of queries instead of a percentage. We set the model's GR to reflect the full (all 2,853 queries) query set's link-to-NIL ratio, and explore the incurred effect on a system with P, R and F1 values equal to our system in case the modeled GR's size equals our GR's (Set 1), and in case it contains only 1,000 queries (Set 2). The results are shown in Table 3. This shows, e.g., that removing 10 queries out of 2,853 barely has any influence on the resulting performance measures (Set1 Max $-$ Min = 0.02), whilst it can already change the performance measures on the smaller query set by almost a full percent (Set2 Max $-$ Min = 0.08). This suggests usage of our model to estimate the possible reduction of the uncertainty on performance metric that can be obtained from extra query annotations, leaving it to the user whether or not this estimated reduction suffices to justify the needed extra annotation effort.

3.3 System Comparison

To track the influence of the query set size on its ability to distinguish between two systems, we compared the top run for the TAC 2013 EL task to each best run from all other 26 teams. This we did for increasing query set size, ranging from only a few queries to the entire query set. For a given query set size, using the TAC queries in their original (arbitrary) order up to that rank, we determined for each of those 26 systems whether we can reject the hypothesis that the top system is not better than the considered system in terms of F1 measure, in a one-tailed test at the $p = 0.05$ confidence level. This was obtained from the difference in F1 between both systems, for 1,000 bootstrap samples created from the considered query subset. The results of this experiment are depicted in Fig. 4, displaying the number of systems which are not significantly less effective

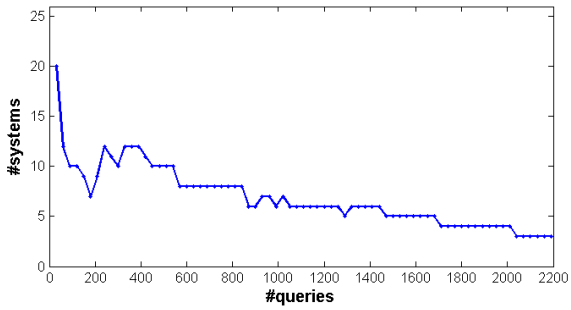

Figure 4: Number of systems not significantly less effective than the top scoring system.

than the top system, for increasing query set size. The figure clearly shows that increasing a query set's size unmistakeably adds to its discriminating power. Yet even for the full query set, a few systems cannot significantly be distinguished from the top system.

4. CONCLUSION

In this paper, we introduced two distinct ways of automatically selecting test queries for the evaluation of Entity Linking systems, demonstrating one method to result mainly in the creation of queries referring to popular entities, and the other to dig more into the long-tail mentions which refer either to lesser known knowledge base entities, or to entities unknown to the knowledge base. We showed the important impact of both methods on system performance, highly suggesting there is no "Golden" query set that unequivocally satisfies all evaluation needs, but instead that the creation of a (useful) query set is dependent on the intended application. We further suggested a method for judging whether annotations for extra queries are useful or not, based on estimating the possible impact on assessed system performance of removing a well defined number of queries, and demonstrated that the size of the query set impacts its ability to significantly distinguish between different systems.

5. ACKNOWLEDGMENTS

This research was carried out in the BEAMER and STEAMER projects, facilitated by the iMinds Media Innovation Center (MiX), and financed by the Flemish Agency for Innovation by Science and Technology (IWT).

6. REFERENCES

[1] R. Bunescu and M. Paşca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, pages 9–16, 2006.

[2] B. Carterette and I. Soboroff. The effect of assessor error on ir system evaluation. In *SIGIR*, pages 539–546, 2010.

[3] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP*, pages 708–716, 2007.

[4] T. Demeester, R. Aly, D. Hiemstra, D. Nguyen, D. Trieschnigg, and C. Develder. Exploiting user disagreement for web search evaluation: an experimental approach. In *WSDM*, 2014.

[5] L. Mertens, T. Demeester, J. Deleu, and C. Develder. Ugent participation in the tac 2013 entity-linking task. In *Proc. 6th Text Analysis Conference (TAC 2013)*, 2013.

[6] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.