Using @Twitter Conventions to Improve #LOD-based Named Entity Disambiguation

Genevieve Gorrell, Johann Petrak, and Kalina Bontcheva

Department of Computer Science, University of Sheffield,
211 Portobello, Sheffield, UK
Initial.Surname@sheffield.ac.uk

Abstract. State-of-the-art named entity disambiguation approaches tend to perform poorly on social media content, and microblogs in particular. Tweets are processed individually and the richer, microblog-specific context is largely ignored. This paper focuses specifically on quantifying the impact on entity disambiguation performance when readily available contextual information is included from URL content, hash tag definitions, and Twitter user profiles. In particular, including URL content significantly improves performance. Similarly, user profile information for @mentions improves recall by over 10% with no adverse impact on precision. We also share a new corpus of tweets, which have been hand-annotated with DBpedia URIs, with high inter-annotator agreement.

1 Introduction

A large body of research has focused on Linked Open Data-based Named Entity Disambiguation (NED), where names mentioned in text are linked to URIs in Linked Open Data (LOD) resources (e.g., [18, 11]).

State-of-the-art LOD-based NED approaches (see Section 2) have been developed and evaluated predominantly on news articles and other carefully written, longer texts [23, 5]. As discussed in Section 2, very few microblog corpora annotated with LOD URIs exist and they are also small and incomplete.

Moreover, where researchers have evaluated microblog NED, e.g. [8], state-of-the-art approaches have shown poor performance, due the limited context, linguistic noise, and use of emoticons, abbreviations and hashtags. Each microblog post is treated in isolation, without taking into account the wider available context. In particular, only tweet text tends to be processed, even though the complete tweet JSON object also includes author profile data (full name, optional location, profile text, and web page). Around 26% of all tweets also contain URLs [4], 16.6% – hashtags, and 54.8% – at least one user name mention.

Our novel contribution lies in systematically investigating the impact that such additional context has on LOD-based entity disambiguation in tweets (see Section 6). In particular, in the case of hashtags, tweet content is enriched with hashtag definitions, which are retrieved automatically from the web. Similarly, tweets containing @mentions are enriched with the textual information from that Twitter profile. In the case of URLs, the corresponding web content is included

as context. Disambiguation performance is measured both when such context expansion is performed *individually* (i.e. only hashtags, only URLs, etc.), as well as when all this contextual information is used *jointly*.

A new corpus of around 800 tweets is made available, annotated with DBpedia URIs, by multiple experts (Section 3). The tweets contain hashtags, URLs, and user mentions, including many with corresponding DBpedia URIs (e.g. @eonenergyuk). The resulting dataset[1] is split into equally sized training and evaluation parts.

2 Related Work

There are a number of openly available, state-of-the-art LOD-based NED systems (for a complete list see [5]), including DBpedia Spotlight [18], AIDA [11], and, most recently, AGDISTIS [27]. Another notable example is TagMe, which was designed specifically for annotating short texts with respect to Wikipedia [9]. A comparative evaluation of all openly available state-of-the-art approaches, except the most recent AGDISTIS, is reported in [5], using several available news datasets, which however exhibit very different characteristics to social media.

Microblog named entity disambiguation is a relatively new, under-explored task. Recent tweet-focused evaluations uncovered problems in using state-of-the-art NED approaches in this genre [1, 8], largely due to the brevity of tweets (140 characters). There has been limited research on analysing Twitter hashtags and annotating them with DBpedia entries, to assist semantic search over microblog content, e.g. [16]. NER systems targeted at microblog text don't commonly utilize these cues, for example treating hashtags as common words, e.g. [15, 21] or not considering them, as in TwiNER [14]. Shen et al. [26] use additional tweets from a user's timeline to find user-specific topics and use those to improve the disambiguation. Huang et al. [13] present an extension of graph-based disambiguation which introduces "Meta Paths" that represent context from other tweets through shared hash tags, authors, or mentions. Gattani et al. [10] make use of URL expansion and use context derived from tweets by the same author and containing the same hashtag, but don't evaluate the contribution of this context to end performance, and don't make use of hashtag definitions or user biographies.

Microblog corpora created specifically for LOD-based entity disambiguation are very limited. Some, e.g. Ritter's [24], contain only entity types, whereas those from the MSM challenges [25, 3] have anonymised the URLs and user name mentions, which makes them unsuitable for our experiments. Corpora created for semantic linking, such as Meij [17], are not well suited for evaluating named entity disambiguation, since annotations in those corpora include entities which are not mentioned explicitly, as well as generic concepts (e.g. art).

[1] Available from https://gate.ac.uk/applications/yodie.html

3 The Annotated Tweet Corpus

A set of 794 tweets were collected. 400 of those were tweets from 2013 coming from financial institutions and news outlets, which were chosen due to the relatively high frequency of named entities within. They are challenging for entity recognition and disambiguation, since capitalisation is not informative (all words have initial capital), but on the other hand, they are quite grammatical.

The rest are random tweets collected in 2014, as part of the DecarboNet project on analysing online climate change debates [22]. Keywords such as "climate change", "earth hour", "energy", and "fracking" were used and the 394 tweets were chosen as a representative sample, containing sufficient named entities, without significant repetition.

The 794 tweets (see Table 1) were annotated manually by a team of 10 NLP researchers, using a CrowdFlower interface. Each tweet was tagged by three annotators, chosen at random by CrowdFlower amongst these ten. Annotations for which no clear decision was made were adjudicated by a fourth expert, who had not previously seen the tweets. Unanimous inter-annotator agreement occurred for 89% of entities, which can be used as the upper bound on performance attainable by an automatic method on this dataset and task.

While others [12] have used automatic named entity recognition tools to identify entities and only then carry out manual disambiguation, we avoided bias by first asking annotators to manually tag all tweets with named entities. Then entity disambiguation annotation was carried out in a second manual annotation round, where annotators had to choose amongst one of the candidate URIs or NIL (no target entity), when no target entity exists in DBpedia. The latter case is quite frequent in tweets, where people often refer to friends and family.

Highly ambiguous entity mentions (e.g. Paris), however, can have tens or even over a hundred possible candidate DBpedia URIs. Since showing so many options to a human annotator is not feasible, instead, during data preparation, candidate entity URIs were ranked according to their Wikipedia commonness score [19] and only the top 8 were shown, in addition to "none of the above" and "not an entity" (to allow for errors in the entity tagging stage).

	Tweets	Total NEs	URLs	Hashtags	@mentions
Total	794	681	504 (236)	359 (188)	334 (316)
Training	397	257	242 (112)	172 (88)	167 (157)
Test	397	424	262 (124)	187 (100)	167 (159)

Table 1: Corpus Statistics

The resulting corpus contains 252 person annotations, 309 location annotations, 347 organization annotations and 218 nil annotations. With respect to URLs, user mentions, and hashtags, Table 1 shows the statistics of their availability in the corpus. The number in brackets shows how frequently expanded context can be obtained for them. It is evident that whilst URLs appear frequently in the data, only around half of them are successfully retrieved. This is due to both web pages becoming outdated and also URLs often being trun-

cated in re-tweets where tweet character limits are often exceeded. Similar to the findings of earlier studies, hashtags are less frequent, and again, we are able to retrieve their definitions from the web automatically in only half of the cases. @mentions are the least frequent; however, we were able to obtain the corresponding Twitter user profiles for most of them, with variable quality.

4 The NED Framework

In order to experiment with the effects of tweet expansion on NED performance, and in particular, on how additional contextual information impacts different semantic similarity metrics (see Section 5), we make use of a NED framework built on top of GATE [6], called YODIE [2]. It combines GATE's existing ANNIE NER system with a number of widely used URI candidate selection strategies, similarity metrics, and a machine learning model for entity disambiguation, which determines the best candidate URI.

In this section, we provide a brief overview of YODIE, focusing in particular on the similarity metrics investigated in the tweet expansion experiments in this paper, and the final disambiguation stage, since these are the parts that are influenced by tweet expansion. For a complete description, including more information about candidate selection and the features used for disambiguation see [2]. We conclude the section with a comparison positioning YODIE with respect to other state-of-the-art NED systems, which demonstrates that YODIE is a representative framework in performance terms in which to conduct our experiments.

4.1 Scoring and Feature Creation

At each NE location and for every candidate, YODIE calculates a number of normalized scores, which reflect the semantic similarity between the entity referred to by the candidate and the context of its mention:

- *Relatedness Score*: introduced in [20], uses the proportion of incoming links that overlap in the Wikipedia graph to favour congruent candidate choices.
- *LOD-based Similarity Score*: similar to above but based on the number of relations between each pair of URIs in the DBpedia graph (introduced next).
- *Text-based Similarity Scores*: measure the similarity between the textual context of the mentioned named entity and text associated with each candidate URI for that mention (see below).

LOD-based Similarity Scores: LOD-based similarity scores are calculated as the number of direct or indirect relations between each candidate URI of an ambiguous named entity and the URIs of candidates for other named entities within a given context window. All relations present in DBpedia are considered for this calculation. We calculate several separate scores, for the number of direct

[2] https://gate.ac.uk/applications/yodie.html

relations ($a \rightarrow b, a \leftarrow b$) between URIs a and b, and for the indirect relations between a and b that involve one other node x ($a \leftarrow x \rightarrow b$, $a \rightarrow x \leftarrow b$, $a \rightarrow x \rightarrow b$, $a \leftarrow x \leftarrow b$).

For example, if the document mentions both *Paris* and *France*, a direct relations score is assigned to `db:Paris`, as the two are connected directly via the `db:country` property. On the other hand, if *Paris* appears in the context of *USA*, a higher indirect score for `db:Paris,_Texas` will be assigned by combining the DBpedia knowledge that Paris, Texas is related to Texas and the additional knowledge that Texas is a US state.

Since any NE mention can have several candidate URIs, and each of the other entities in the context can have several candidates too, YODIE calculates the value of each score as the sum over all pairs for each candidate, divided by the distance in characters between the candidate locations [2]. This means that where a relationship is found, both candidates in question will benefit. The combined LOD-based similarity score is a sum of scores for all relation types each weighted by the inverse square of the degrees of separation, i.e., indirect relations receive a quarter weighting compared with direct relationships. The context for the calculation of these scores and the relatedness score is set to 100 characters to the left and 100 to the right of each location, rounded down to the nearest whole word, as a heuristic designed to make the calculation quickly achievable by reducing the number of neighbours.

Text-based Similarity Scores: YODIE's text-based similarity scores evaluate candidate URIs on the basis of how well the surrounding context matches representative text associated with the candidate URI. Three approaches to text-based similarity are supported, as follows:

1. Text from the URI's DBpedia abstract, limited to the words within the first 5000 characters, again, as a heuristic to avoid very variable computation times due to unexpectedly large documents.
2. The abstract text as above, plus the literals from all datatype properties for the URI.
3. All previous words, plus the literals from all datatype properties of directly linked other URIs.

The entire tweet is used as context, subject to stop word removal and lowercasing. The three textual similarity scores for each candidate URI are calculated as the cosine similarities between the context vector and the vector of the respective text for the candidate. Cosine was chosen for its wide popularity.

4.2 Disambiguation

As described above, YODIE generates a number of similarity scores, each providing different information about the fit of each candidate URI to the entity mention. The process of deciding how to combine these scores to select the best candidate URI is non-trivial. YODIE uses LibSVM[3] to select the best candidate.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

A probabilistic SVM is used, in order to make use of the classification probability estimates in selecting a candidate. Default parameters are used, since tuning failed to improve performance.

Training data for the model consists of one training instance for each candidate generated by the system on the training corpus. Each instance receives a target of `true` if the candidate is the correct disambiguation target and `false` otherwise. The values of the various similarity metrics are used as features (see [2] for details). This means that at application time, the model assigns to each candidate a class of true or false, along with a probability. This classification is independent of the other candidates on that entity, but ranking of the candidate list is able to be performed on the basis of the probability. The most probable URI is thus assigned as the target disambiguation for this entity, unless its probability is below a given confidence threshold, in which case "nil" is assigned. We trained on TAC KBP data from 2009 to 2013, excluding the 2010 set [4], along with the AIDA training set [11], and the tweet training set introduced in section 3.

4.3 Comparison to Other NED Systems

In order to validate YODIE as a framework suitable for performing the tweet expansion experiments, we compare performance with other available state-of-the-art NED approaches. Results are reported on the widely used "Test B" part of the Aida/CoNLL corpus [11] (see Table 2). This corpus contains 231 documents with 4485 target annotations.

The results shown in Table 2 for AGDISTIS (the most recent NED system) are those reported in [27]. AIDA [11], Spotlight [18, 7], and TagMe [9] results are as reported in [5] (this AIDA result is indicated with a "2013" suffix in the table). The latter paper also includes a detailed comparison against the Illinois Wikifier and Wikipedia Miner. However, due to space limitations here, these worse performing systems are excluded. The results for the latest Aida algorithm in 2014 [12] are also included, based on a local installation of the 2014-08-02 version of the system (as recommended on the AIDA web page) and the 2014-01-02v2 version of the dataset[5]. Results for several other widely used NED services are also included (default parameter settings are used), namely Lupedia[6], TextRazor[7] and Zemanta[8].

As can be seen in Table 2, on this news dataset, YODIE performs second best. The latest AIDA system outperforms others by some margin on the AIDA dataset, but amongst others, YODIE compares favourably. The rest of the paper will focus on more in-depth experiments and analysis of the various tweet expansion techniques and their impact on NED precision and recall.

[4] http://www.nist.gov/tac/2013/KBP/

[5] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/`

[6] `http://lupedia.ontotext.com/`

[7] `https://www.textrazor.com/`

[8] `http://www.zemanta.com/`

System	Prec.	Recall	F1
YODIE	0.62	0.65	0.64
Aida/2013	**0.74**	0.34	0.47
Aida/2014	0.70	**0.74**	**0.72**
Spotlight	0.31	0.40	0.35
TagMe	0.61	0.56	0.58
AGDISTIS	0.64	0.56	0.60
Lupedia	0.58	0.31	0.40
TextRazor	0.35	0.58	0.34
Zemanta	0.51	0.29	0.37

Table 2: AIDA B Evaluation

5 Expansions Studied

This section describes our methodology for retrieving and utilizing expanded context from hashtags, user mentions, and URLs.

5.1 Performing Tweet Context Expansion

In the YODIE NED framework, each individual tweet is represented as a separate document. Context expansion is performed by temporarily including additional text about each @mention, hashtag and URL link. Subsequent stages then process the original tweet text together with each individual context section. This approach makes use of the flexible way in which GATE models arbitrary text spans in documents, through stand-off annotations. In other words, processing can be restricted to just those parts of the expanded tweet which are of interest, e.g. the original tweet text and the context created from all definitions of all hashtags present in the tweet. Metadata features on the annotations are also used to establish the link between the original hashtag, @mention, or URL in the tweet and their corresponding expansion text. Since documents in GATE are dynamically editable, all additional content is removed, after NED processing is completed and before evaluation.

Figure 1 illustrates an expanded tweet (yellow highlighted text in the main pane); "KAGAWA will be allowed to rejoin Borussia Dortmund in January in a swap deal which would see defender @NSubotic4 join #MUFC http://tiny.cc/4t19ux". The tweet includes a hashtag, `#MUFC`, highlighted in blue, a user ID, "NSubotic4" in pink and a URL in green. Each of these items is expanded into the longer section of correspondingly coloured text included below, in the order they appear in the tweet. Entities are indicated in a darker shade.

Expansion of Hashtags: Hashtags are a Twitter convention, which makes it easy for users to find all tweets on a given topic or event, e.g. a name (`#obama`), an abbreviation (`#gop`), concatenations of several words (`#foodporn`). Some hashtags are also ambiguous, i.e. can have different meanings at different times. Since many hashtags contain entity mentions, which are often missed by state-of-the-art NED systems, we experimented with expanding tweets with hashtag definitions, provided by the web site `https://tagdef.com`

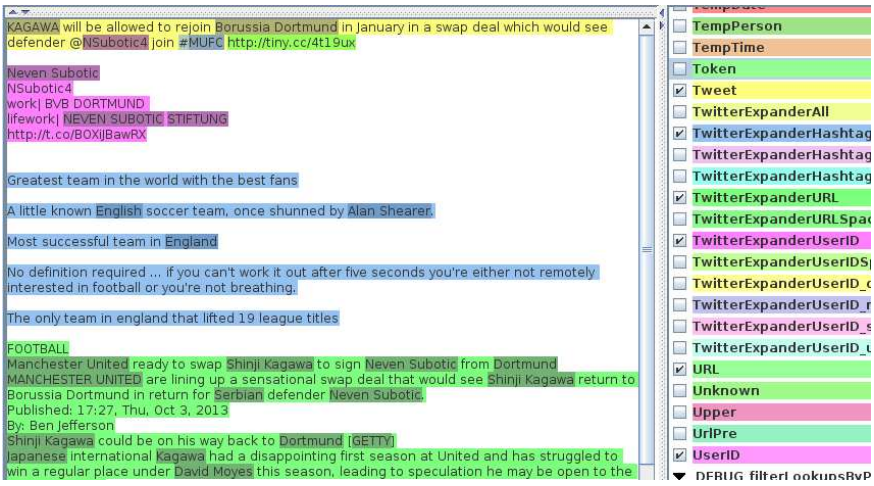

Fig. 1: A screenshot showing tweet expansions and entities found in them.

Tagdef hashtag definitions are crowdsourced. Since anyone is free to enter any definition, there is plenty of noise, for example in the expansions for Manchester United Football Club in our screenshot example, the expansions are humorous and opinionated rather than informative. The website offers an API that returns up to 6 definitions, which are all added as additional context to the original tweet document. TagDef does not have definitions for all hashtags: in the 794 documents there are 359 hashtags of which 171 have no definitions.

Expansion of @Mentions: For each @mention, tweets are enriched with the following user profile textual information: `name`, `screen_name`, `location`, `description`, and `url`. The latter are not expanded with more content recursively. GATE annotations are added which identify from which of these fields the text originates. Not all @mentions found in a tweet can be expanded since the user may have deleted their account or an account may have been suspended. There are 334 @mentions in our corpus, of which 18 could not be resolved.

Expansion of URLs: For each URL in the tweet, the content of the corresponding web page is retrieved and added to the document. Since many web pages contains boilerplate text (e.g. navigational menus), this is filtered automatically and only the core text is added as additional context. Images are currently ignored. Since many people post images in their tweets, this is one of the reasons why URL expansion is not always possible. In addition, the target page may no longer exist or may not be accessible at retrieval time. In our corpus there are 504 URLs, of which 236 could not be expanded.

5.2 Making Use of the Expanded Content

As discussed above, for each candidate URI YODIE calculates a number of similarity scores, which are then used as features in the entity disambiguation model. Our experiments in tweet enrichment focus on its influence via the three

introduced earlier, as well as the possibility of adding new candidates via back-projection of entities, as outlined below.

Contextual Similarity and Expansion: Contextual similarity uses the additional information in the expanded tweets, in order to calculate more reliable textual similarity scores. It treats the newly expanded text as though it were collocated with the original hashtag, user mention, or URL. Thus, if an item of expandable content appears within the context window, the entire, corresponding expanded content is included. Where multiple expansions apply, these are simply added in, since the context vectors are bag-of-words based. In the screenshot example, all of Neven Subotic's twitter profile, all of the hashtag expansions for Manchester United Football Club and the entire content of the URL are included as context for the entities in the tweet.

LOD-based and Relatedness Similarity Scores and Expansion: Since semantic relations between two candidate URIs are often sparse, we experimented also with using entities from the expanded context, in order to overcome this. As before, the entire expanded content is treated as though it were collocated with the item it is an expansion of. This means that relation-based similarities are calculated not only between candidates for the target entity and other candidate entities in the context window, but also between the target candidates and candidates for entities in the expanded content. In the screenshot example, we can see for example that "Alan Shearer" appears as an entity in the hashtag expansion, so in calculating a LOD-based similarity score for Borussia Dortmund's candidates, we consider whether they are related to candidates for Alan Shearer.

Back-Projection of Entities In addition, we experiment with *improving entity disambiguation recall*, based on @mention expansions. This is motivated by the fact that an @mention may directly represent an entity which should be linked to the knowledge base, e.g. in our example, `@NSubotic4`. However, the concrete user name often does not get recognized as a named entity and therefore no candidate URIs are generated for it. Nevertheless, textual user names from the tweet author profiles often get recognized as named entities. In the example, the name "Neven Subotic" appears twice as a recognized entity in the expansion.

Therefore, we experiment with projecting the information from the named entity recognized in the expanded user profile, back on to the original @mention, thus potentially finding entity candidates which would not have been identified otherwise. In effect, this assigns the full list of candidate URIs from the named entity onto the user mention. This new candidate list is then used by YODIE for context-based disambiguation.

6 Experimental Results

This section presents the experimental results that demonstrate the impact of the three tweet expansion techniques on the semantic similarity scores discussed above. Statistics are presented on the entire tweet corpus, since scoring takes place before the ML-based disambiguation. This allows us to evaluate how the

scores change without reserving training data for ML. Using a larger corpus maximizes the reliability and informativeness of the results. We present evaluation on the conditions outlined below, chosen for their interest and informativeness, since the full set of combinations would be large.

- **Base** The baseline condition includes no tweet expansion at all. Where machine learning is used, no tweet expansion was used in the training data.
- **Id** @mention expansions are used only. Where machine learning is used, only @mention expansions are included in the training data.
- **Url** URL expansions are used only. Where machine learning is used, only text from URL content is included in the training data.
- **Hash** Hashtag expansions only are used. Where machine learning is used, only hashtag expansions are included in the training data.
- **Id+Proj** @mention expansions are used along with back-projection of entities found in the expansion to create an entity on the @mention where previously there was none. Back-projection without @mention expansion doesn't make sense, hence these conditions must be evaluated together. This type of expansion is used only in the machine learning training data.
- **All** This experiment includes all expansions.
- **Id+Proj+Url** We explicitly evaluate the combination of Id+Proj and URL expansions without hashtags for reasons that become apparent in section 6.2.

Finally, having considered how Twitter expansions affect the LOD-based and contextual similarity scores separately, we consider the impact on the entire system, i.e. including the ML-based disambiguation stage. See Section 6.2 for final system performances in comparison to other state-of-the-art systems. The results are reported on the test corpus, the training corpus having been used along with the TAC and AIDA corpora to train the support vector machine disambiguation model.

6.1 Impact of Tweet Expansion on Individual Similarity Features

There are several ways in which tweet expansions can influence YODIE's performance; via their influence on each of the of scores, and via the creation of new candidates via back-projection from @mentions. These also influence the decisions made by the disambiguation SVM model, hence impact of tweet expansion on the similarity scores is investigated first here, independent of the particular disambiguation algorithm.

In order to give an idea of the contribution of each score, results are reported for precision, recall and F1, obtained where the best candidate is selected on the basis of that individual score alone. It should be noted that naturally results obtained from such individual scores are comparatively low, since overall performance is made possible only by several features being used in combination by the ML model.

Therefore, to put these individual scores in context, consider that if we select, for each entity, a URI from the candidate list at random, an F1 measure of 0.229

is achieved on the test tweet corpus. If we select the best ranked candidate URI based on URI frequencies in Wikipedia, this achieves an F1 of 0.521. URI frequency in Wikipedia, intuitively, indicates how important an entity target is in general world knowledge. This turns out to be a very hard baseline to beat, since not only is the most common candidate more likely to be correct by definition, but also it is more likely to be mentioned in the corpus. These two scores therefore demonstrate the range of performance realistically achievable by a metric, giving a lower (F1=0.229) and upper bound (F1=0.521). The performance of the three individual similarity metrics examined here, as expected, falls within this range.

	Prec.	Recall	F1
Base	**0.416**	0.267	0.326
Id	0.399	**0.276**	0.326
Url	0.414	0.272	**0.328**
Hash	0.385	0.253	0.305
Id+Proj	0.318	0.266	0.313
Id+Proj+Url	0.373	0.269	0.312
All	0.373	0.260	0.306

Table 3: LOD-based Sim. Score

	Prec.	Recall	F1
Base	0.236	0.244	0.240
Id	**0.253**	0.272	**0.262**
Url	0.236	0.242	0.239
Hash	0.235	0.241	0.238
Id+Proj	0.244	0.269	0.256
Id+Proj+Url	0.249	**0.276**	**0.262**
All	0.250	0.266	0.258

Table 4: Relatedness Score

As can be seen in Tables 3 and 4, the Wikipedia relatedness score and the LOD-based similarity score respond slightly differently to the inclusion of tweet expansion information. In particular, the expansion of user ids benefits recall for LOD-based similarity (table 3), but also decreases precision. The best overall F1 is achieved with URL expansion only, but even then the score is not much higher than the F1=0.326 without any expansion. Results where back projection and @mention expansions are performed are substantially worse.

In contrast, the relatedness similarity score (Table 4) does benefit substantially from the additional textual information. @mention expansions, in particular, lead to improvements in both precision and recall, whereas hashtag and URL expansions have relatively little impact.

For text-based similarity on DBpedia abstracts alone, table 5 shows that @mention expansion and entity projection lead to improved precision and recall, with hashtag and URL expansions bringing limited benefit only.

When other textual fields from DBpedia are used in addition to abstracts, as context for comparison of each candidate, then tweet expansion leads to even higher performance gains (see table 6). In this case, URL and hashtag expansions both lead to improved results, with further gains brought by back projection and @mention expansion. The best overall result is when all expansions are combined.

	Prec.	Recall	F1
Base	0.201	0.421	0.272
Id	0.208	0.436	0.282
Url	0.194	0.407	0.263
Hash	0.204	0.427	0.276
Id+Proj	**0.217**	**0.463**	**0.295**
Id+Proj+Url	0.212	0.454	0.289
All	0.216	0.461	0.294

Table 5: Text: Abstracts Only

	Prec.	Recall	F1
Base	0.221	0.379	0.279
Id	0.226	0.389	0.286
Url	0.234	0.402	0.296
Hash	0.234	0.401	0.295
Id+Proj	0.235	0.414	0.300
Id+Proj+Url	0.247	0.434	0.315
All	**0.253**	**0.446**	**0.323**

Table 6: Text: Abstracts Plus

6.2 Impact on Overall Disambiguation Performance

The full impact of tweet expansion on NED performance was also evaluated. Table 7 shows the results, where the three similarity features discussed individually in the previous section are now used in combination by the SVM disambiguation model. All other YODIE features and parameters remain unchanged.

We can see that in terms of F1, the biggest improvement comes from @mention expansions including also back-projection of entities. Compared with the baseline, the difference in accuracy is significant to $p < 0.0001$, as established using the McNemar Sign Test. The contribution of @mention expansion alone is not significantly better than the baseline. Hashtags, however, do produce a significant improvement in accuracy ($p = 0.046$), as do URLs ($p = 0.021$).

The confidence threshold on the disambiguation probability produced by the SVM is tuned on the dataset where all expansions are carried out. This leads to the levelling effect across precision and recall that we see in the final result. When compared against the other models, using all tweet expansion strategies leads to a slightly lower recall, but higher precision. Ultimately, this leads to the best overall performance in terms of F1 score.

Since hashtags contribute only marginally, we also examine whether this expansion could be excluded without impact on overall performance. Therefore, results were calculated using @mention with back-projection and URL expansion only (see row "Id+Proj+Url"). This leads to higher accuracy, compared to the system that includes all expansions. F1, however, is higher where hashtag expansion is included. The improvement in disambiguation accuracy is significant at $p = 0.004$; however, depending on the application, a higher F1 might be preferable. The difference in F1 can not be assessed for significance using a paired test.

We also evaluated whether hashtag and URL expansion could be excluded, since the difference in accuracy between all three expansion strategies versus including only @mention expansion with back-projection ("Id+Proj") is not statistically significant ($p = 0.1441$). However, when overall disambiguation accuracy with added URL expansion ("Id+Proj+Url") is compared against "Id+Proj" alone, the latter is indeed significantly worse ($p = 0.011$). Coupled with the fact that F1 also decreases, the conclusion is that URL expansion helps, when used in combination with the two @mention expansion strategies.

The relative contribution of the three types of context expansion cannot be predicted easily for a different corpus, since the distribution of hashtags, user mentions, and URLs can vary from one tweet dataset to another. Nevertheless, extrapolating on the basis that if accuracy improvement due to an expansion type is x, and we had n successful expansions of that type in the corpus, then the improvement per successful expansion is x/n. Therefore, for a hypothetical corpus of 397 documents containing one single successful @mention expansion per document, we might see an accuracy improvement of 0.23; for URL expansions, 0.06; and for hashtag expansions, 0.04. The actual value in real terms, however, depends on the likelihood of those expansion types occuring in an actual corpus and the likelihood that expanded contextual information will be available at disambiguation time.

	Prec.	Recall	F1	Acc.
Base	0.442	0.550	0.490	0.550
Id	0.444	0.557	0.494	0.557
Id+Proj	0.444	0.642	0.525	0.642
Url	0.452	0.568	0.504	0.568
Hash	0.446	0.559	0.496	0.559
Id+Pr+Url	0.452	**0.660**	0.536	**0.660**
All	**0.495**	0.623	**0.552**	0.623

Table 7: Overall Result

System	Prec.	Recall	F1
YODIE (Base)	0.44	0.55	0.49
YODIE (Exp)	0.50	0.62	**0.55**
Aida 2014	**0.59**	0.38	0.46
Lupedia	0.50	0.24	0.32
Spotlight	0.09	0.51	0.15
TagMe	0.10	**0.67**	0.17
TextRazor	0.19	0.44	0.26
Zemanta	0.48	0.56	0.52

Table 8: Tweet Comparison

6.3 Contextualizing Potential Performance Gain

In order to contextualize the magnitude of improvement obtained within results obtained by state-of-the-art NED methods, YODIE's performance with and without tweet expansion is compared on the evaluation part of the tweet corpus described in section 3. The best performing systems obtain F1 scores in the range of 0.46 to 0.52, as table 8 shows, so the six point gain in F1 that we have shown to be possible through the use of tweet expansion is substantial, and sufficient to reposition a system in comparison with others.

7 Conclusions and Future Work

This paper investigated the impact on named entity disambiguation in tweets, when the original tweet text is enriched with additional contextual information from URLs, hashtags, and @mentions. The tweet expansion approaches investigated here can easily be incorporated within other LOD-based NED approaches, through the integration of the relatedness, textual similarity, and LOD-based similarity scores.

Our experiments demonstrated that tweet expansions lead to significantly improved NED performance on microblog content. In particular, overall accuracy improves by 7.3 percentage points, an improvement of 13.3% compared with the

baseline. Performance gain is slightly lower for F1 – an improvement of 6.2 percentage points (11.3% over the baseline).

The main gains arise from the ability to disambiguate @mentions in which the tweet-text only baseline fails to identify their DBpedia referent. The dominant contribution in this case, therefore, is in terms of recall. It should also be noted that even without mention expansions, URL and hashtag expansions also lead to statistically significant improvements.

Limitations to the work include its dependence on the particular candidate scoring metrics and final disambiguation strategy used, since these constitute the channels through which tweet expansion can impact on performance. Future work will involve evaluating tweet expansion in the context of other systems in order to further investigate this interaction.

Acknowledgements The authors wish to thank all volunteers from the NLP research group in Sheffield, who annotated the tweet corpus. This work was partially supported by the European Union under grant agreements No. 287863 TrendMiner and No. 610829 DecarboNet, as well as UK EPSRC grant No. EP/I004327/1.

References

1. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Semantic enrichment of Twitter posts for user profile construction on the social web. In: ESWC (2). pp. 375–389 (2011)
2. Aswani, N., Gorrell, G., Bontcheva, K., Petrak, J.: Multilingual, ontology-based information extraction from stream media - v2. Tech. Rep. D2.3.2, TrendMiner Project Deliverable (2013), http://www.trendminer-project.eu/images/d2.3.2_final.pdf
3. Basave, A.E.C., Rizzo, G., Varga, A., Rowe, M., Stankovic, M., Dadzie, A.S.: Making sense of microposts (#microposts2014) named entity extraction & linking challenge. In: 4th Workshop on Making Sense of Microposts (#Microposts2014) (2014)
4. Carter, S., Weerkamp, W., Tsagkias, E.: Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. Language Resources and Evaluation Journal (2013)
5. Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 249–260. WWW '13 (2013)
6. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with gate's full lifecycle open source text analytics. PLoS Computational Biology 9(2) (2013)
7. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proc. of the 9th Int. Conf. on Semantic Systems. pp. 121–124. I-SEMANTICS '13, New York, NY, USA (2013)
8. Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Bontcheva, K.: Analysis of named entity recognition and linking for tweets. Information Processing and Management 51, 32–49 (2015)
9. Ferragina, P., Scaiella, U.: Fast and accurate annotation of short texts with wikipedia pages. IEEE Software 29(1), 70–75 (2012)

10. Gattani, A., Lamba, D.S., Garera, N., Tiwari, M., Chai, X., Das, S., Subramaniam, S., Rajaraman, A., Harinarayan, V., Doan, A.: Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. Proceedings of the VLDB Endowment 6(11), 1126–1137 (2013)
11. Hoffart, J., Yosef, M.A., Bordino, I., Furstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Conf. on Emp. Methods in Nat. Lang. Processing. pp. 782–792 (2011)
12. Hoffart, J., Altun, Y., Weikum, G.: Discovering emerging entities with ambiguous names. In: Proc. of the 23rd Int. Conf. on World Wide Web. pp. 385–396 (2014)
13. Huang, H., Cao, Y., Huang, X., Ji, H., Lin, C.Y.: Collective tweet wikification based on semi-supervised graph regularization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. pp. 380–390 (2014)
14. Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B.S.: Twiner: named entity recognition in targeted twitter stream. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 721–730. ACM (2012)
15. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 359–367 (2011)
16. Lösch, U., Müller, D.: Mapping microblog posts to encyclopedia articles. Lecture Notes in Informatics 192(150) (2011)
17. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: Proc. of the Fifth Int. Conf. on Web Search and Data Mining (WSDM) (2012)
18. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding light on the web of documents. In: Proc. of I-SEMANTICS. pp. 1–8 (2011)
19. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proc. of the 17th Conf. on Information and Knowledge Management (CIKM). pp. 509–518 (2008)
20. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: In Proceedings of AAAI 2008 (2008)
21. Murnane, E.L., Haslhofer, B., Lagoze, C.: Reslve: leveraging user interest to improve entity disambiguation on short text. In: Proceedings of the 22nd international conference on World Wide Web companion. pp. 1275–1284. International World Wide Web Conferences Steering Committee (2013)
22. Piccolo, L.S.G., Alani, H., De Liddo, A., Baranauskas, C.: Motivating online engagement and debates on energy consumption. In: Proceedings of the 2014 ACM Conference on Web Science (2014)
23. Rao, D., McNamee, P., Dredze, M.: Entity linking: Finding extracted entities in a knowledge base. In: Multi-source, Multi-lingual Inf. Extraction and Summarization. Springer (2013)
24. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: An experimental study. In: Proc. of EMNLP (2011)
25. Rowe, M., Stankovic, M., Dadzie, A., Nunes, B., Cano, A.: Making sense of microposts (#msm2013): Big things come in small packages. In: Proceedings of the WWW Conference - Workshops (2013)
26. Shen, W., Wang, J., Luo, P., Wang, M.: Linking named entities in tweets with knowledge base via user interest modeling. In: Proc. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. pp. 68–76. ACM (2013)
27. Usbeck, R., Ngonga Ngomo, A.C., Auer, S., Gerber, D., Both, A.: Agdistis - graph-based disambiguation of named entities using linked data. In: International Semantic Web Conference (2014)