UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Bringing microblog updates updates to Wikipedia

by

Maël Thomas

A project report submitted for the award of
MSc Web Technology

Supervisor: Dr Leslie Carr
Examiner: Dr Nicholas Gibbins

September 19, 2012

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

A project report submitted for the award of MSc Web Technology

BRINGING MICROBLOG UPDATES UPDATES TO WIKIPEDIA

by Maël Thomas

Wikipedia is not to be expected to display real-time information. One cannot but notice that this would indeed be convenient, for its articles are obvious reference pages for each subject and provide detailed context around that information. We present Weeki, an application that extracts meaning from Twitter to add a real-time layer to Wikipedia. This project aims to bridge the gap between these two Web information giants, through a complex information retrieval step followed by a carefull integration that adds a complementary dimension to both reading and editing the online encyclopedia. This paper provides a detailed design of the application features, and a proof of concept implementation.

Contents

List of Tables

Acknowledgements

I would like to thank my supervisor and second examiner for their help and interesting suggestions. A special thank as well to Camellia sinensis.

Introduction

Twitter and Wikipedia are two of the most visited websites, both enjoying wide recognition respectively as a social network and an online and collaborative encyclopedia. While Wikipedia presents semi-structured and high quality content, Twitter exemplifies the real-time web thanks to its straightforwad message model. We consider them in this work as complementary information sources.

While Wikipedia's information value is now unquestionable[1], the automated extraction of meaning from Twitter is a complex task, recently illustrated and increasingly subject of attention for the research community.

This project first focuses on the extraction of information from individual tweets and from the microblogging network as a whole, their linking to Wikipedia articles as well as their encoding as semantic data. But more importantly, we give our vision of a concrete integration of this stream to Wikipedia, designing an application and providing a technical proof of concept.

Our application is built with a Web page enrichment approach, to assure a circumspect integration to a website that has built its success on strong guidelines.

We believe that this work provides new and desirable features to the online encyclopedia, and moreover draws new possibilities for user contribution. Though not viable in its current state, we hope that our work will be followed or drive similiar initiatives.

This paper is organised as follows. The background section gives the context, and introduces relevant natural language tools as well as important related works. The design of our idea is then presented, followed by a description of our implementation. We finish with a critical evaluation and indicate interesting future directions for this project.

[1] Although of course not perfect and often benchmarked against traditional encyclopedias

Chapter 1

Background

1.1 Context

At the center of this project are Wikipedia, the reference online free and collaboratively edited encyclopedia,[1] and Twitter, the microblogging patform that has established itself as a major player on the social Web. Wikipedia's content is well categorized, and its articles describe concepts of reference for the Web, extracted to form what can be viewed as a core for the network of Linked Data, DBpedia. Twitter rode the wave of the social web, attracting users and information sources (blogs, news agencies...) that would make it a full-fledged information medium. It is an immensely vast, varied and above all real-time source of, alas, unstructured and noisy information, its scope and message model bringing great challenges to the task of information extraction: tweets can convey unrestricted subjects, provide little context and are written in an often informal style. No measures have been officilly provided or adopted for a formalization of writing, except for hashtags, that can be as irregular as the text message itself.

Thus, a complementarity can be observed between a Wikipedia that lacks real-time updates and a Twitter that provides only unstructured data.

Twitter and Wikipedia are obviously built on fundamentaly different principles. Those principles are in fact what makes the two platforms so pertinent for their usage : Twitter is extremely easy to use, everything being done to be able to tweet from everywhere with one or two steps, allowing for astonishing network

[1] https://en.wikipedia.org/wiki/Wikipedia

dynamics. On the contrary Wikipedia edition and reading asks for more time and more focus, leading to a highly trustable source of information.

Objections to our work will legitimately evoque the danger of denaturing the two services by altering and mixing up these foundations, which are reasons for their popularity and recognition. A tweet classification feature would impact Twitter's wish of maximal simplicity; providing real-time updates to Wikipedia would be a door open to content of a weaker quality. Exigences of neutrality must be reflected upon, statements must be correctly referenced with trusted sources, writing style must be elaborate.

Providing a more structured model for every tweet is up to the company itself, and would deeply question the principles that made it so popular. We believe however that bringing real time information to Wikipedia is feasible and desirable, with a particular attention dedicated to ensuring that its spirit is kept. Our work does not automatically modify the articles sources, but rather lays a well identified real-time layer on top of them, carefully considers issues of impartiality and puts the last decision for information storage in the hands of the community.

Still growing at more than 8500 articles per day in 2012, it is however less known that the number of active editors on Wikipedia (80 500) represent less than 0,02% of its total monthly unique visitor count (450 million) and that this number of new editors is declining.[2] These statistics may be interpreted in different ways. A positive one might be that the growth of Wikipedia has reached its limit, for the reason that it already describes a near satisfying amount of knowledge about our world in its current state. Another would be that resource constraints are starting to hinder it.[3] These interpretations however do not imply that there is no way to alter this trend. Specifically, we believe that novel features and design could automatically enrich or assist edition, leading to a wider audience of users and editors, drawing on the activity of Twitter's claimed 500 million users through their 400 million tweets per day in 2012, benefiting from the spread of mobile devices.[4]

Interaction between those two gigantic information worlds remains feable. Though Twitter is a subject appreciated by researchers, few of the publications aim at

[2] Statistics for Wikimedia, July 2012; an active editor is a registered user with more than 5 edits per month; http://stats.wikimedia.org/

[3] http://www.guardian.co.uk/technology/2009/aug/12/wikipedia-deletionist-inclusionist

[4] active users have been reported however to represent less than one third of that number http://techcrunch.com/2012/07/31/twitter-may-have-500m-users-but-only-170m-are-active-75-on-twitters-own-clients/, http://news.cnet.com/8301-1023_3-57448388-93/twitter-hits-400-million-tweets-per-day-mostly-mobile/

constructing web applications that would enhance both, and most of the end-user attempts did it one way. For example, Xu and Oard (2011) aims to enhance Twitter search through a topical clustering that uses Wikipedia's information.

To the best of our knowledge, our work is unique as an initiative to conciliate the advantages of both, and more particularly by its focus on the retrieval of relevant tweets *for* Wikipedia, rather than the contrary. However, the first part of this project consists more of an intelligent mash up of existant technologies and services rather than a fundamentally new algorithmic model, and as such, special attention has been given to the *integration* part, where general guidelines are set, elaborate experiments described and demonstrated for some with a basic implementation.

1.2 Related Work

1.2.1 Tweet processing

We need to link tweets to Wikipedia articles, and derive meaning from them for further applications.

1.2.1.1 Entity Linking

In order to attain the objective of this project, a necessary step is to find a way to automatically establish links between tweets and wikipedia articles. These links are semantic, they should make a relation between the concepts that a user may want to refer to in her 140 character microblog post and the concept materialized by a wikipedia article. A natural way to do that is through the approach called annotation: enriching text with links to topics it refers to. Annotation is covered extensively by academic research, as we will see below. Some of those papers particularly focus on the annotation of short text fragments, which further corresponds to our ambition to treat tweets.

Annotation of text can be categorized in terms of the type of output that is produced. Though other annotation types exist, we are primarily looking for a step focused on entity linking, to directly link tweets to Wikipedia concepts.

The process of knowledge extraction that associates Wikipedia concepts to ngrams of a text input, usually called wikification, is a particularly focused variation to

Named Entity Recognition (NER) of great interest to us. Several wikification engines have been released, we will review the most relevant ones that have been documented. Relatively recent, they exploit the size and quality of Wikipedia, relying specifically on the large link structure carefuly selected and reviewed by humans. In fact, most of the wikification systems rely on Wikipedia's internal anchor set, an anchor being a portion of text that has been attributed a hyperlink to another Wikipedia page.

The simplest approach to annotate text is to match fragments (word n-grams) with concept names, here the Wikipedia article titles. This *lexical matching*, of course, does not perform very well, as seen in the table 3 of Meij et al. (2012), yielding poor recall scores because of the title-based only query as well as poor precision due to the absence of a step to disambiguate concepts, explained below. The first wification system reported is Wikify Mihalcea and Csomai (2007). It relies on two steps, detection and disambiguation. Detection computes the *link probability* score of each n-gram in the input text, and keeps those higher than a given threshold.

Equation.

The disambiguation step, performed on each n-gram for which an anchor has been found, aims to select the most relevant concept among the list of concepts pointed through this anchor. The approach taken by Wikify authors is to compare local and global context (global refers to the entire text to annotate) with training data from Wikipedia itself.

The authors of WikiMiner, Milne and Witten (2008b) have then built their system on this first achievement and added new scoring functions. They released the software and a set of several Web services.[5] Following the approach of Bunescu et al. (2008), their disambiguation step mainly relies on two scores, *commonness* and *relatedness*, that will be employed as features for machine learning.

Commonnness, also called prior probability, is defined as follows :

Equation

Relatedness is a disambiguation criteria that selects the candidate concept that resembles most the input text's unambiguous annotations (anchors that point to one concept only). The similarity measure between two concepts relies on the

[5] http://wikipedia-miner.cms.waikato.ac.nz/demos/annotate/

amount of incoming links that the two corresponding Wikipedia pages have in common, as defined in one of their previous works, Milne and Witten (2008a).

Equation

Finally, the detection phase relies on the measure of link probablility and machine learning.

Both Wikify and WikiMiner, as well as the work of Chakrabarti et al. (2009), are designed for annotating long and rich texts, such as encyclopedic articles or news reports. The TAGME algorithm Ferragina and Scaiella (2012) aims to address the challenge of wikification on short input texts, and with a complexity that allows for real-time processing. It is also based on two steps, disambiguation and pruning.

Disambiguation can be performed as a global or local scale. Local disambiguation, though quicker, does not achieve better results than the global counterpart Narr et al. (2011). TAGME's *disambiguation* step differs from WikiMiner's in that it takes into account the "vote" of all the other potential annotations, not only those that are unambiguous, in order to compensate for the much limited context that a short text holds.

Equation.

An optimum is then found between the relatedness and commonness scores to associate a concept to each anchor.

Equation.

The *pruning* step computes two features, the *link probability* and the *coherence*, and judges an annotation as irrelevant if their average is lower than a precise threshold.

Coherence Equation Pruning criteria

A last set of algorithms was proposed very recently to annotate microblog posts Meij et al. (2012), CMNS-RF. The additional machine learning step (RF), yielding significant improvements over the TAGME algorithm (see table 8), involves however a complex combination of features, and the first step (Commonness), though still evaluated as an improvement over TAGME, has been on the contrary previously proven weaker Ferragina and Scaiella (2012).

As described in a benchmark by the authors of Meij et al. (2012) [6], there exist several other wikification algorithms, among which are Yahoo! content analysis, Wikimeta or DBpedia Spotlight. This comparison confirms our conviction that TAGME is adapted to our project. Hence its set of algorithms has been retained as the base for our work, as described in the following chapter link.

It can be noted that the Google Web anchor dictionary released and documented in Spitkovsky and Chang (2012) gives hope of an extension of the anchor database that powers all wikification algorithms cited above. Although recall would certainly benefit from this, without a severe filtering step precision would be dramatically reduced, as the database contains all the hazardous anchor texts written on the entire Web.

1.2.1.2 Information extraction from micro posts

Although the wikification step already creates links between Wikipedia articles and tweets, sufficient to provide a stream for some of the applications envisioned and implemented in this project, there is much more to be extracted from the messages using information extraction tools, that eventually enable the construction of semantic assertions. We focus particularly on open information extraction approaches, a subject that has received much attention lately, as they are more adapted to the scale of the Web, and its diversity of domains.

The range of Named Entity Recognition tools (NER), to which Wikification belongs, is interesting to us. Some of them only perform the task of attributing predefined types to entities, revealing for example that "Paris" is an entity of type "Location". NER systems have reached very satisfying performance, but run into difficulties when applied to short and less formal text inputs, and particularly in the case of tweets. Recent research efforts have however focused on this task, yielding significant improvements Ritter et al. (2011), demonstrated on the concrete case of event extraction, cited below. NER systems can be used as a complement to wikification, which can cast a shadow on important entities not (yet) included in Wikipedia but that should be considered as subjects in assertions, as explained by Lin et al. (2012). Nakashole and Weikum (2012) advocate this "Dynamic Entity Recognition".

Word sense disambiguation (WSD) algorithms precisely perform the task of entity linking. Rusu et al. (2011) atempted to adapt two WSD algorithms to annotate

[6] edgar.meij.pro/comparison-semantic-labeling-algorithms-twitter-data/

text with entities from Linked Open Data sets, resolving this issue of missing or inadequate Wikipedia entries.

Another set of tools are dedicated to relation extraction, extracting the link between entities, as a first step to obtain semantic predicates. The ReVerb algorithm Fader et al. (2011), one of several works by the University of Washington's AI department[7], relies on the OpenNLP library [8] and improves precision and recall as compared to the standard extractors, TextRunner by Etzioni et al. (2008) and WOE by Wu and Weld (2010). Subsequent work has again significantly increased performance over ReVerb Mausam et al. (2012), with an extraction not limited to verbs but also nouns and adjectives, the addition of contextual information, with its software released on the Web.[9]. Other works have focused on the extraction of n-ary relations Akbik and Löser (2012).

Narr et al. (2011) have an interesting proposal, considering that everything contained in a tweet that holds an entity is an annotation of that entity (objective, subjective statements, sentiment...). Their first step is normalization Han and Baldwin (2011) of the irregular words of the tweet.

Research has also been conducted by Rizzo et al. (2012) on the unification of NER tools (NERD) and the translation of the results to RDF (NIF). Ngomo and Heino (2011) unify the output of more diverse NLP tools using neural networks.

Nakashole and Weikum (2012) emphasize on the need for real-time extraction algorithms to produce up to date assertions, also relying on the couple named entity recognition - relation extraction.

Gerber and Ngomo (2012) built the BOA framework to infer triples from unstructured Web content by mapping natural language patterns to the predicates of existing Linked Data. Aside from creating new sets of Linked Data, the system exposes this database of correspondence between predicates and natural language constructs.

Very recent works have tackled the global problem of inferring RDF/OWL linked data from unstructured text: Augenstein et al. (2012) and Presutti et al. (2012)[10]. Both works coordinate a series of NLP tools, that includes NER, C&C and Boxer that output Discourse Representation Structures, and WSD. These entities and relations are then published in the form of semantic triples.

[7] http://ai.cs.washington.edu/projects/open-information-extraction
[8] http://opennlp.apache.org/
[9] available at https://github.com/rbart/ollie
[10] This second work will be presented at EKAW 2012 and is not available at the moment

1.2.1.3 Network-wide information extraction

Along with the growth of Twitter as a popular and unique network has come research efforts on its analysis, starting with Kwak et al. (2010). The new field of information extraction from Twitter as a whole is of particular interest to us, a step above the complex retrieval of pertinent information from individual "micro" inputs of text, involving classifying and filtering the dynamics of the network.

Earthquake detection gives a good example of a working and very useful information extraction process Sakaki et al. (2010). Twitter trending topics (described as being topics that are "immediately popular"[11]) have been watched and compared to traditional news headlines in other media. Kwak et al. (2010) shows that more than half of these can be categorized as "timely breaking" or persistent news (p.7).

Zhao et al. (2011) perform a deeper analysis of the relation between Twitter and traditional news media. Findings particularly relevant to our project are that their coverage of the range of categories is similar, that there are more tweets on personal life and pop culture than on world events, that tweets can convey further information about brands and celebrities, and finally that retweets are a way to find important trendy topics.

As a particularly tinted source of information, the integration of tweets to Wikipedia would have to be rigorously framed to respect the principles evoked earlier. On the other side of the coin, it could certainly emerge as a valuable addition, bringing an "opinion layer" with information holding a focus on personal concerns, different perspectives through unique points of view on real-world events as described by Diakopoulos et al. (2010), a diversified coverage brought by any user speaking about unpopular subjects or very local events Becker et al. (2011).

Research has also been conducted on the detection of news. It includes using similarity measures to group similar tweets in news stories Phuvipadawat and Murata (2010) or discover interesting tweets and reliable tweeters by mining first the news on traditional websites Sankaranarayanan et al. (2009). The distinction between real-world events and Twitter-centric events has also been performed successfuly Becker et al. (2011). Or more generally, a broad classification of tweets relying on the study of user intentions that usually follow specific patterns Sriram et al. (2010).

[11] https://support.twitter.com/articles/101125-about-trends

Real world events bring up a high level of excitement on the network–political conventions, launch of new devices..., sometimes raising the frequency of tweets to astonishing topical rates (e.g 2012 american election conventions reached 53 000 tweets per minute [12]). Ritter et al. (2011) have studied the extraction, aggregation and classification of these events, and released an impressive Web app demonstrating their work with an up to date calendar of incoming events.[13]

Twitter has also been praised for transporting information more quickly than conventional media, at least for the particular case of celibrity deaths Sankaranarayanan et al. (2009), a characteristic that could make our initiative even more relevant in the aim of bringing live information updates to Wikipedia.

1.2.2 Semantic Microblogging

Research publications have attempted to bring semantic Web concepts to microblogging. SMOB is a platform for open, semantic and distributed microblogging Passant et al. (2010), that would, with wide adoption, definitely be of great interest as a source for our system. Other projects aim at semantifying tweets : HyperTwitter defines a syntax to enable users to make assertions (equivalence, specialization, known and new semantic predicates) directly in tweets, and a service that extracts them to an RDF graph Hepp (2010); TwitLogic reviews several of those proposed "nanoformat" syntaxes, introducing a new and more natural one and focusing on RDF encoding (using known vocabularies : FOAF, SIOC...) and publishing principles Shinavier (2010); the Linked Open Social Signal initiative defines content encoding as well, real-time SPARQL querying, streaming, and push notifications Mendes et al. (2010). Although these publications do not help us for the information extraction step given the low adoption of the syntaxes defined, the semantic frameworks they define will help us for encoding and publishing the processed tweets.

1.2.3 Linked Data

In 2007 an extension was developped to bring semantic authoring to MediaWiki Krotzsch et al. (2007), but was not adopted by Wikipedia[14]. A project called

[12] http://blog.twitter.com/2012/09/dnc2012-night-3-obamas-speech-sets.html
[13] http://statuscalendar.cs.washington.edu/
[14] Scalability reasons were evoked, but adoption of a new and more demanding syntax by the community could also have been a reason

Shortipedia was then launched to independently capture semantic assertions from all over the Web Vrandečić et al. (2011). This has in some ways led to Wikidata, adopted as a new project by the Wikimedia foundation and described as a semantic knowledge base for the world.

The last proposal of this project is to leverage both Twitter and Wikipedia to populate this promising triple repository. DBpedia would obviously be the link between the encyclopedia and Wikidata **?**.

The aim of Han and Baldwin (2011) is similar to ours in their objective of mining information from twitter and publishing it as a semantic database, but their objectives remain vague : "use the DBpedia ontology representation and augment it with additional required structures".

1.2.4 Web page enrichment

One of the aims of this project is to automatically annotate Wikipedia pages via tweets and the information extracted from them. This topic of Web page enrichment regularly surfaces in research projects and commercial applications. In fact, it has a long story, that includes the research on link injection in the context of applying the concepts of open hypermedia to the Web Carr et al. (1998) Carr et al. (2001), and as mentionned by Milne and Witten (2008b), simpler applications of annotation algorithms, with Microsoft's Smart-Tag service to automatically add links to text displayed by Internet Explorer, and Google's AutoLink toolbar feature.[15] This acknowledgement of the need to annotate the Web was shared by W3C's Annotea project Kahan and Koivunen (2001) and has been recently implemented by new companies, creating for example the blog tool Zemanta[16] and a collaborative annotator iGlue[17], followed as well by news corporations as the BBC Kobilarov et al. (2009).

The approach of Web page annotation can be salutary as a highly formal semantic structure could be felt as too restrictive for authors Millard et al. (2005), which could explain the adoption issue of Semantic MediaWiki evoked above.

Extensions and widgets have been created for MediaWiki[18], but they embed tweets in very simple ways that mirror the official twitter add-ons (e.g. a simple user feed).

[15] Both initiatives were discontinued due to impartiality concerns coming from the public
[16] http://www.zemanta.com/
[17] http://www.iGlue.com/
[18] The software that powers Wikipedia http://www.mediawiki.org

Chapter 2

The Weeki system

In this chapter we present our work on the design and implementation of the application.

2.1 Engine

We aim for a high recall retrieval of tweets, and the highest quality of enrichment.

2.1.1 Retrieval

Two possible approaches Our objective in this step is to obtain a real-time list of tweets for each article browsed. Harnessing the principles of wikification, there were two approaches that could lead to our expectations.

The first one, that could be called "catch them all", operates as follows: given a concept (that is a wikipedia article page), we want to retrieve real-time tweets that implicitely or explicitely refer to it. Annotation methods would simply solve the problem in the case that we could retrieve in real-time all the tweets published on the network: they would be annotated, resulting in a relational table linking tweets to identified concepts, that could easily be reversed to serve on-demand the tweets relating to one concept. Unfortunately, retrieving all the tweets is not something that we can afford for a couple of inescapable limitations in the context of a summer project. Available technologies can now process such a big amount of data: the software stack that we had chosen included a high performance HTTP

client by Twitter itself, Finagle, a NoSQL database and a language known for its scalability (also used by Twitter), Scala[1]. The main limitations to this approach, however, simply came from the Twitter streaming API[2]. First, only 1% of the flow can be retrieved, leaving most of the information aside. Then came the idea to retrieve more tweets by spliting the retrieval with the filter streaming endpoint, but it turns out that the API allows only one simultaneous connection. Official data retailers would have given us access, for a fee, to the entire feed, a load that would have conducted us to switch to a distributed model such as Hadoop,[3] requiring significant engineering efforts (to define the jobs), setup, and computing resources.

2.1.2 Linking tweets to a Wikipedia article

Hence the need for a shift in our approach, to take the problem in the opposite way: the processes have to be fired on a Wikipedia article request, making calls for the tweets it needs only, without any global treatment or database at this stage[4]. Tweets that relate to a concept have to be selectively retrieved using the search API. As a consequence, we need to grasp the principles of the wikification algorithms and adapt them to serve our proper objective.

The first step consists of knowing how to query the Twitter search API, a request that would be made for every article served to users. The easiest way is to define the query as the article's title. However, given the varying writing styles of the messages on the network and possible synonyms of this title, many tweets would be missed.[5] Following Ferragina and Scaiella (2012) (paragraph 4), an anchor dictionary is built using normalized article titles, redirect titles, removing single numbers (e.g. years), single letters and anchors whose link frequency is too low. This step can be described as a task of query expansion, as it adds to our query string word synonyms and alternative grammatically correct writing forms. The addition of lexical variations as described by Ritter et al. (2011) (paragraph 2.1) would take care of the grammar errors or voluntary abreviations often found in tweets.

[1] http://twitter.github.com/finagle/, http://redis.io/, http://www.scala-lang.org/

[2] https://dev.twitter.com/docs/streaming-apis/streams/public

[3] http://hadoop.apache.org/

[4] A final implementation with several users would of course need a cache for each article

[5] It must be noted that Twitter obviously does not only look for search keywords in tweets. Details about this search algorithm are nevertheless unknown.

Previous to the annotation step is the process of tokenization, that structures the input text as a sequence of words and treats the particular features of twitter (e.g. replacing user @ mentions by user names, removing the retweet RT keywords, removing the # of hashtags).

The wikification algorithm described in depth in X is applied to all the tweets retrieved. This additional step is intended to prune the irrelevant tweets catched by this expanded search, by simply dropping those whose wikification output does not include the original Wikipedia concept. Of course, applying the TAGME algorithm described in paragraph X can be judged as cumbersome for this pruning task. The advantage, however, it to to fully annotate at the same time the tweets kept as relevant with their concepts, moving us ahead to our semantic enrichment step, for linking to Wikipedia concepts at the same time annotates the tweets with DBpedia entities. The algorithm is besides designed for low computing times.

2.1.3 Stream Enrichment

Links embedded in a tweet are not to be ignored, as they are in some cases its main value, the surrounding text being a description or opinion expressed about them. In fact, according to Narr et al. (2011) 44% of tweets contain urls. A web service is used to resolve the shortened links and retrieve essential information about them.[6] As a large proportion of them are articles (newspapers, blogs...), we are particularly interested in getting their title, abstract, tags and image, added as a property of the annotated tweet object. The actual use for these links is to turn them into article references as explained in paragraph, but it could be interesting in future work to go further and perform a deeper analysis of the linked page. It must also be noted that Twitter has recently released a new feature, expanded tweets, that lets partner accounts include a link preview directly in the tweet.[7] This operation, similar to our link resolution, is at the moment rather elementary, but could in the future take a very important role on the network. Twitter has in fact by the past announced a new feature called annotations, that would let users embed any kind of metadata in tweets, including microformats and links to Wikipedia concepts, but has not been released yet.[8]

Twitter hashtags are basically treated as simple words by the wikification step, since users can omit to include words in a tweet when a hashtag is sufficient to set

[6] Diffbot, free for a limited number of monthly calls http://www.diffbot.com/
[7] http://blog.twitter.com/2012/06/experience-more-with-expanded-tweets.html
[8] www.readwriteweb.com/archives/what_twitter_annotations_mean.php

its context. These hashtags (initials for example) might not however be linked by wikification algorithms if they do not exist as anchors in Wikipedia. The works of Lösch and Müller (2011) and Laniado and Mika (2010) enable us to go further by linking hashtags to Wikipedia concepts. This step comes however to the price of higher computation time, and it usefulness could be questioned given the fact that those hashtags not recognized by wikification algorithms often refer to in-network discussions that could hardly be linked even manually to any Wikipedia concept.

The classification of tweets into different categories, namely news, events, opinions, deals, and private messages, is interesting to us for two reasons. First, it enables us to filter this last category, which does not correspond to the objectives of this project. This helps us clear out the large portion of tweets that consists of daily and personnal chatter[9]. Secondly, this labeling of tweets will be used for our user interface paragraph.

Finally, the identification of news stories and events using the methods described in paragraph enable us to make a specific usage of these tweets, that can be essential side informations about concepts and are usually interesting to a large audience. The algorithms described previously can however require the entire database of tweets for a given period of time. This luxury that we could not afford (see paragraph) would have to be reconsidered, or if possible, their methods adapted to treat only the stream of tweets retrieved by our engine.

Natural language processing tools as well as frameworks combining them, detailed in the paragraph, enable us to derive semantic assertions (RDF triples) from individual tweets. These systems would have to be tailored to the specificity of our input, for instance by using a wikification adapted to tweets, TAGME, instead of the Wikipedia Miner module in Augenstein et al. (2012). We aim at collecting assertions that include Wikipedia entities, but could link them to other sets in the fashion of the interlinking of DBpedia[10]. Possible overlapping steps in these off-the-shelf algorithms would have to be examined in order to ensure optimal computing time.

[9] Although the Wikification step already filters messages that do not contain a reference to a Wikipedia concept

[10] http://wiki.dbpedia.org/Interlinking

2.2 Application Design

The description of the application that we make of this stream of tweets provided by the engine is structured as a list of *features*, that happen on top of the Wikipedia article pages. In order to respect a desire to read exclusively the original content of a Wikipedia page, or to enjoy a distraction free experience, a slider button would be provided to each user to permanently deactivate these features.

2.2.1 Page Enrichment

2.2.1.1 Smart Tweet Insertion

We designed an integration of tweets on the Wikipedia article based on rules of concept matching. Tweets are inserted in well identified *spots*, areas of the page where they are judged relevant.

The easiest spots to match are the existing internal links embedded in the article text, that represent concepts. All the various section and subsection endings are also candidate spots, with the matching performed on their headers. Their end is targeted since they are usually written in a chronological order, while tweets almost exclusively convey recent information.

When a match is detected, ponctual spots (existing links) are filled with a small icon, reacting to a user click by displaying the tweet in a small popup. Area spots such as sections are directly followed by a "related tweets" link that would, on click, unfold a horizontal list of tweets. These insertions must remain discrete enough not to perturb the reader that just wants to focus on the encyclopedic content of Wikipedia.

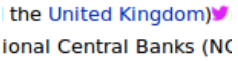

Figure 2.1: A ponctual spot filled with a tweet marker (We apologize for altering the Twitter logo, but it helped make the markers more obvious for this experiment) (implemented feature)

As explained in paragraph, tweets would come with a label (News, Event, Opinion) that would define the background color of messages as displayed on the page, allowing users to know what kind of information they should expect from them.

Figure 2.2: The corresponding tweet rolled out on click (implemented feature)

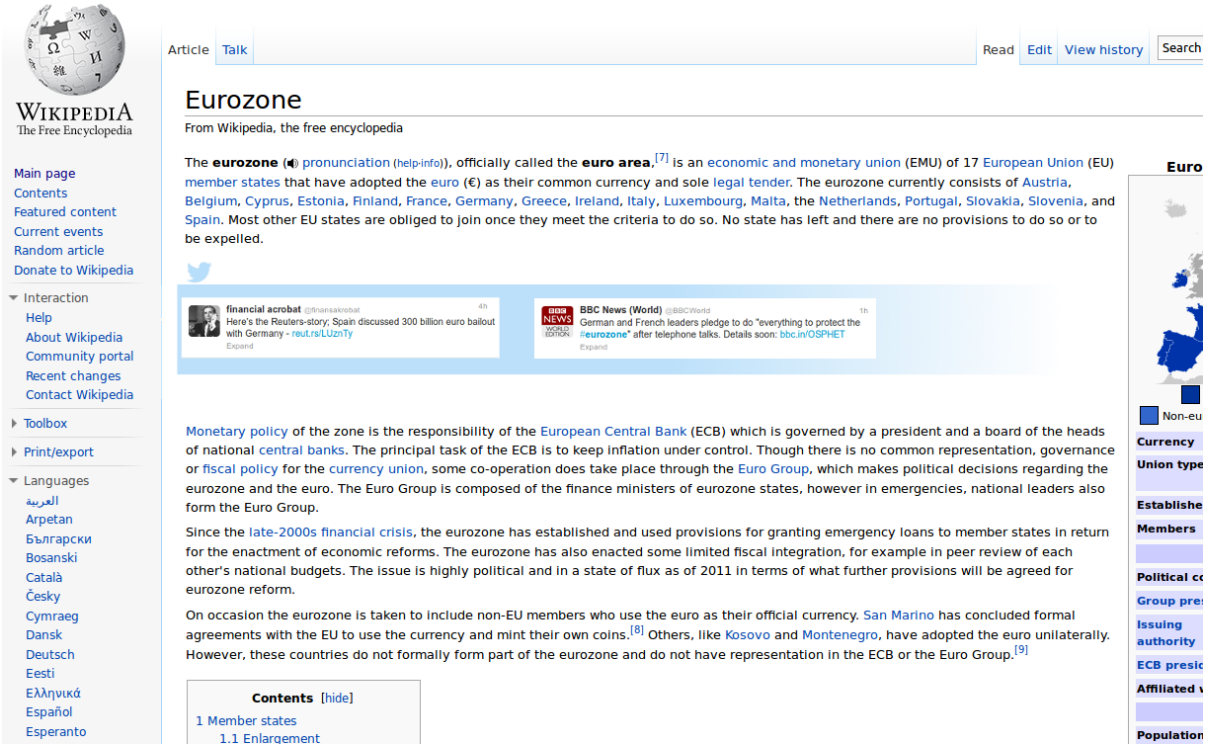

Figure 2.3: Area insertion (mockup)

Moreover, our project further plans on an interaction with users reading the page, allowing them to act on these inserted tweets. Actions would be mapped to evocative buttons such as delete, which lets users judge the insertion as irrelevant, move to a more fitted spot, and finally store, for tweets that are meaningfull enough to be permanently included in the article's tweet database or directly as Wikipedia references (e.g. a tweet that holds value in itself as an original declaration, as the famous https://twitter.com/BarackObama/status/241392153148915712).

2.2.1.2 Reference suggestion

Another implemented feature is an automatic and real-time suggestion of references to users that are updating the article. Their text input is processed with our set of algorithms in the same manner as tweets, leading to eventual matches between extracted concepts. Links extracted from tweets, based on rules that define them as acceptable, are suggested below the input form along with an "add reference" button. This feature, if carefully implemented, could help users back up their claims more easily, taking advantage for example of the innumerable trustworthy news article links published on Twitter.

The wealth of information extracted through natural language processing tools paragraph could in the same manner serve for a match between tweets and article claims that have been marked as needing a citation, poviding for example a drop-down list of links identified as potential references.[11]

2.2.1.3 Headline Suggestion

Research has shown how meaningful information can be extracted from Twitter as a dynamic network paragraph Work. Particularly, we aim to use the extracted breaking news stories and identified events that revolve around the concept through the display of an informative live banner right after the article's title. To prevent the possibility of false positives as well as correct but secondary items to be displayed, we again rely on the individual judgments of the large Wikipedia community as safeguards to act on these automatic headlines.

When reading an article on Wikipedia, it usually takes some time before finding the paragraph where the most recent information is written. This headline suggestion feature would enable acknowledgement of important news about this resource at first sight. An illustrative use case would be about athletes, whose latest competition performances reported in the news are of capital interest, especially at the time of competitions, but may not be instantly updated in the article text.

2.2.2 Linked Databases Population

The last feature defined in this project aims to employ the Wikipedia community to judge assertions sourced from the twitter stream. Users would be proposed

[11] http://en.wikipedia.org/wiki/Wikipedia:Citation_needed#Citation_needed

unobstrusive popups asking them to confirm wether assertions appear to them as both correct and worth of attention. These assertions could comprise multiple disambiguation candidates that users could choose from if needed.

Figure

This feature could eventually be used as a way to populate the Wikidata project. In order to guarantee precision, a filter could be designed to consider only assertions whose predicates include properties listed in the DBpedia ontology[12]. Other restrictions could be defined in order to respect the future Wikidata policies, for example people involved in this process could be restricted to registered Wikipedia users.[13]

The practical feasibility of this feature would obviously depend on the performance of the tools used upstream to infer semantic assertions, but the main bet of our implementation nonetheless resides in this semi-automated process involving human judgement as an ultimate step.

Another interesting application of this feature would be to release the set of mappings that have been found between natural language relations and ontology predicates, similar to the BOA base described by Gerber and Ngomo (2012). The high confidence of these mappings that were validated by users would give way to multiple applications in the domain of information extraction from the Twitter network as well as others websites sharing this unformal text style.

2.3 Implementation

Given that this project is largely based on the design of a novel application, an early implementation was welcome as a proof of concept. The infrastructure of Weeki, as well as two of the Wikipedia integration features described in paragraph have been implemented: smart tweet insertion and reference suggestion. However, the engine consists only of the wikification step. This implementation is available on a Git repository[14].

[12] http://dbpedia.org/Ontology
[13] http://meta.wikimedia.org/wiki/Wikidata/Preventing_unwanted_edits
[14] https://github.com/laem

2.3.1 Software Stack

Following the novel nature of the application, modern technologies have been employed for its implementation.

- The programming language chosen is Scala. At the same time improving on and compatible with Java,[15] it is an adequate choice for scalability and extensibility. Brand new Web frameworks have been developped on top of it, and it is definitely adapted to complex Web applications.[16] Its functional style also allows for more elegant and readable programming. Last but not least, it has been chosen by Twitter for a number of critical back-end processes[17].
- The application is built using the Play! 2.0 framework. Now natively in Scala, it facilitates asynchronous web programming. Play was documented and usable enough (easy setup, well known MVC model, automatic code reloading) for a one month deployment.
- Websocket is a new technology providing a communication channel between browsers and servers, compatible with modern browsers.[18] It advantageously replaces the AJAX/Comet duo in our case, and is straightforward to use with the Play framework.
- The client is coded in the incontrovertible Javascript and jQuery.[19]

2.3.2 Extensibility

Although this implementation is only a proof of concept previewing our ideas, it is highly extensible for future development.

Classes can simply be added to the back-end code to, for example, implement the text to RDF feature. They could be algorithms coded in Scala or Java, or simply Web service clients coded with the Play! helpers. The new data produced by these additions can be passed along one of the existing websocket channels, or simply added to a new one. The client code is written in Javascript, allowing the addition

[15] http://www.infoq.com/news/2009/07/scala-replace-java
[16] Notably with the Akka library http://akka.io/
[17] http://www.readwriteweb.com/hack/2011/07/twitter-java-scala.php
[18] https://en.wikipedia.org/wiki/Websockets
[19] This implies that clients with Javascript disabled would not be able to use our application. The proposed redesign of Wikipedia (see paragraph) also currently falls short on this constraint.

of jQuery libraries to extend functionality and interface, and enabling a future release as a browser extension or jQuery plugin. In order to be fully integrated to Wikipedia, the code would have to be directly included in the MediaWiki project, or in a more modest way as an extension.

2.3.3 Process description

The code logic that constitutes our proof of concept works as follows, triggered on the request of a Wikipedia article. We note that the annotation Web service used is Wikipedia Miner[20] as access to the TAGME service[21] could not be obtained. Asynchronous calls are performed where desirable.

Figure 2.4: A screenshot of the current implementation that has just loaded and displayed tweets

- The Wikipedia HTML page is retrieved and modified to include our CSS files and client-side Javascript code.
- A Twitter search API client perdiodically makes a search request corresponding to the requested article (no query extansion is performed).
- Tweets are annotated with service calls, and their URLs resolved with the Diffbot service.[22]
- These enriched tweets are sent to the client through a Websocket channel, as JSON arrays.

[20] http://wikipedia-miner.cms.waikato.ac.nz:8080/services/?wikify
[21] http://tagme.di.unipi.it/
[22] http://www.diffbot.com/

- The smart tweet insertion feature is performed by matching concepts and appending HTML via jQuery code.
- The reference suggestion demo calls the server's annotation service through a second Websocket channel and displays the corresponding list of extracted URLs.

2.3.3.1 Anchor Dictionary

The anchor dictionary provided by the Wikipedia Miner project suffices for the query expansion task as well as the needs of the wikification algorithm.[23] However, the Wikipedia dump from which this information is extracted dates back to July 2011, which matters given the very recent information conveyed by tweets.

We have thus attempted to build the anchor dictionary ourselves. Wikipedia releases periodic XML dumps of its articles. The task would be to parse each article for internal links, and store the anchor as well as the target article id. We have used the Wikipedia Preprocessor[24], that uses the Perl MediaWiki::DumpFile script, to get an output file listing these anchors. A Scala script lets us create one file for each article with its id as a name, that contains the list of anchors and corresponding count.

This output format makes the query expansion task trivial. However, we ran into two issues. First, as each file takes at least 4.0 ko of space on modern filesystems, the total exceeded by far the disk space given for this project. For performance reasons, the wikification algorithm chosen would eventually need to be implemented, and would require a list of article ids for each anchor found in the input text. This storage method does not allow for an efficient lookup of ids.

Another storage method would hence need to be used to construct the dictionary. NoSQL databases seem to be a good choice given the scale of the task.

[23] http://wikipedia-miner.cms.waikato.ac.nz/services/exploreArticle?id=333355&labels=true and http://wikipedia-miner.cms.waikato.ac.nz/services/search?query=kiwi

[24] http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/

2.4 Project Management

As part of the MSc Web Technology, the successive steps of this project were initially planned according to the following Gantt chart.

Week # / *week beginning :*	**1** 13/6	**2** 20/6	**3** 27/6	**4** 4/7	**5** 11/7	**6** 18/7	**7** 25/7	**8** 1/8	**9** 8/8	**10** 15/8	**11** 22/8	**12** 29/8	**13** 5/9	**14** 12/9	**15** 19/9
Activities and milestones															
Break															
Idea refinement															
Lit review, dev platform choice															
Research															
Implementation															
Writing-up															
Milestone – demonstrate to sup/examiner												*			
Milestone – dissertation draft complete														*	
Final corrections															
Milestone – Hand-in															*

This planning was on overall respected, except for a more important overlap between research and implementation, with comings and goings between specification and implementation.

Chapter 3

Evaluation and future work

3.1 Critical Evaluation

3.1.1 Testing

It is clear that this project is not elaborated enough to be the subject of rigorous technical evaluation. It can nevertheless be planned to serve as guidelines for future work.

The engine of our application that retrieves, annotates and wraps the tweets as semantic objects would have to be tested with Information Retrieval methods.[1] Each component would have to tested against manual annotation of the tweets in input, to be able to compute recall, precision and F1 score. While some of these algorithms have been tested in the corresponding research paper on the case of tweets (e.g. TAGME), a good amount of time would have to be spent on the others (e.g. relation extraction). It is probable that recall may be affected by the fact that the concepts of some tweets are not mentionned and hard to infer. However, we can bet that the redundancy of messages on the network would reduce that loss.

The second part of this work, the application idea, could not be evaluated technically as the questions to answer, such as "how useful it would be ?", are more abstract. We propose an evaluation based on a survey, that would have to be

[1] http://en.wikipedia.org/wiki/Precision_and_recall

conducted when more features would have been implemented, or at least carefully illustrated to avoid "blind" answers reflecting an absence of understanding from a mere description of the functions. A couple of interesting questions are listed hereafter.

- Are you an active Wikipedia contributor (more than 5 edits per month) ?
- Are you an active Twitter user ? If not, do you follow users and read their tweets ?
- How usefull would you judge the smart tweet insertion feature ? The reference suggestion ? The headline suggestion ?
- Would you answer the factual questions (in order to enrich the Wikipedia infoboxes / Wikidata) ?
- How do you judge the global interest of the Weeki application ?

A final testing step would be user experience testing, to evaluate the technical and design aspect of the implementation. Ideally, it would consist of another user survey. Otherwise, a heuristic evaluation adapted to modern Web applications should be applied Thompson and Kemp (2009).

3.1.2 Value and limits of our work

New natural language processing algorithms (NLP) were obviously not developped in this project, nor a framework to unify them for the purpose of our work. NLP applied to tweets is a complex subject of study and a serious contribution to it was not in the scope of this project. However, we did provide a corresponding litterature review that we do hope is exhaustive enough.

The principal value of this work resides in the novel idea introduced–bringing real-time updates on top of Wikipedia through the extraction of meaningfull information from Twitter. We are not aware of similar initiatives that compare to this aspect, excepted those focusing on some parts of it and detailed in the background chapter.

The accompanying implementation has to be seen as a proof of concept to illustrate this idea and elementarily show its feasibility. It however provides a structure ready for future work with a convenient and modern technology stack, and proves to be working, fast, easy to deploy and not requiring any setup (excepted the download of the Play framework).

3.1.3 Reception

Although some users of Wikipedia could judge the addition of real-time updates as undesirable or distracting, we do believe that it would be welcome by a significant proportion of them, particularly by the increasing proportion of active Twitter users that recognize and daily exploit the potential of this network.

This work primarily aims to bring attention on a new information source for Wikipedia as well as ideas to integrate it. Depending on its reception by the Wikipedia community, it could inspire new MediaWiki features (that wouldn't be imposed to users, as explained in paragraph (switch button)), or interesting new browser extensions.

3.2 Future Works

The topic range embraced by this project is large, and several aspects could not be treated in time. Most of them are listed there.

Online Newspapers An interesting area of focus would be online newspapers. They can be used to facilitate the detection of news that matter to Twitter users. The Guardian and the New York Times, among others, have released semantic Web APIs, that could be used in the entity linking step as another source of Linked Data. The general advantage is to incorporate data sets and ontologies that are among the most up to date, to help bridge the gap between real-time but unstructured streams and structured but more static repositories.[2]

[2] http://www.guardian.co.uk/open-platform/blog/linked-data-open-platform?CMP=twt_gu, http://data.nytimes.com/

Content specialization In the tweet query step we have focused exclusively on content. There are however other search parameters relevant to our application: a Wikipedia article about a music festival could in addition trigger the retrieval of tweets posted with a geolocation value comprised in an adequate radius, and display a stream of tweets labeled as local. The assignation of higher weight to tweets posted by specific users (depending on the article) could also be studied.

Personnalized streams User recommendation is a dimension that was not studied in this project. Users that whish to log in with their Twitter account would see the selection of tweets tailored to the characteristics of their account (e.g. accounts they follow). Particular attention would have to be given to the issue of the "filter bubble effect"[3].

Wikipedia Redesign A couple of initiatives have advocated the need for a Wikipedia interface redesign, such as the Athena proposal and the Wikipedia Redefined experiment.[4] These initiatives would bring Wikipedia closer to the graphical and user experience level of modern websites, and we believe that our application would be better integrated and welcome therein.

Additional sources We have focused exclusively on Twitter as an information source, as its success makes it the most useful one, but other microblogging networks should eventually be taken into account, e.g. StatusNet, identi.ca and the new App.net.[5] This could give more diversity, and maybe quality (registered users may post more serious messages than on Twitter).

Generic applicability This project focuses on Wikipedia as a support for the application. The idea of a retrieval, enrichment and integration of tweets relating to a concept, however, could be of great interest for a variety of websites (e.g. Wordpress). It could be interesting indeed to design the framework as generic, with the integration features detailed in this paper incorporated as a component focused on Wikipedia. An API or the ability to create other integration features as extensions to the framework could be imagined.

[3] http://en.wikipedia.org/wiki/Filter_bubble
[4] http://www.mediawiki.org/wiki/Athena, http://www.wikipediaredefined.com/
[5] http://www.status.net/, https://identi.ca/ and http://www.app.net/

Chapter 4

Conclusions

We have in this paper undertaken the task of enriching Wikipedia with meaningful information extracted from the most famous microblogging network.

A review of the state of the art information extraction tools, particularly those more able to deal with the specificities of Twitter messages laid the foundations for a message retrieval and enrichment engine. A particular focus was then set on the concrete design of our application, describing four integration features enabling a richer and live experience for reading and editing Wikipedia articles.

Moreover, an implementation was written to prove the feasibility and the interest of our idea.

We finished with testing suggestions, a critical evaluation of our idea and work, as well as the presentation of several diverse and challenging directions for future work on this project.

This project has been an opportunity to explore the range of research on natural language processing, gain theoritical and practical experience in the design of a rich Web application, as well as for a constant and creative reflection on how to suitably propose novel features for a well established website.

Appendix A

Stuff

The following gets in the way of the text....

References

Akbik, A. and Löser, A. (2012). KrakeN: N-ary Facts in Open Information Extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montr{é}al, Canada. Association for Computational Linguistics.

Augenstein, I., Padó, S., and Rudolph, S. (2012). LODifier: Generating Linked Data from Unstructured Text. In Simperl, E., Cimiano, P., Polleres, A., Corcho, O., and Presutti, V., editors, *ESWC*, volume 7295 of *Lecture Notes in Computer Science*, pages 210–224. Springer.

Becker, H., Naaman, M., and Gravano, L. (2011). Beyond trending topics: Real-world event identification on Twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Bunescu, R., Gabrilovich, E., and Mihalcea, R., editors (2008). *Topic Indexing with {Wikipedia}*, Menlo Park, CA, USA. AAAI.

Carr, L., Hall, W., Bechhofer, S., and Goble, C. (2001). Conceptual Linking: Ontology-based Open Hypermedia. *Proceedings of the 10th international conference on World Wide Web*, pages 334–342.

Carr, L. A., DeRoure, D. C., Davis, H. C., and Hall, W. (1998). Implementing an Open Link Service for the World Wide Web. *World Wide Web Internet And Web Information Systems*, 1(2):61–71.

Chakrabarti, S., Kulkarni, S., Singh, A., and Ramakrishnan, G. (2009). Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 457, New York, New York, USA. ACM Press.

Diakopoulos, N., Naaman, M., and Kivran-Swaine, F. (2010). Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122. IEEE.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ferragina, P. and Scaiella, U. (2012). Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Software*, 29(1):70–75.

Gerber, D. and Ngomo, A.-c. N. (2012). Extracting Multilingual Natural-Language Patterns for RDF Predicates. In *EKAW 2012*.

Han, B. and Baldwin, T. (2011). Lexical Normalisation of Short Text Messages : Makn Sens a # twitter. *Computational Linguistics*, V(212):368–378.

Hepp, M. (2010). HyperTwitter: Collaborative Knowledge Engineering via Twitter Messages. In Cimiano, P. and Pinto, H., editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 451–461. Springer Berlin / Heidelberg.

Kahan, J. and Koivunen, M.-R. (2001). Annotea: an open RDF infrastructure for shared Web annotations . In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 623–632, New York, New York, USA. ACM Press.

Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., and Lee, R. (2009). Media Meets Semantic Web — How the BBC Uses DBpedia and Linked Data to Make Connections. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 723–737, Berlin, Heidelberg. Springer-Verlag.

Krotzsch, M., Vrandecic, D., Volkel, M., Haller, H., and Studer, R. (2007). Semantic Wikipedia. *Web Semantics Science Services and Agents on the World Wide Web*, 5(4):251–261.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 591, New York, New York, USA. ACM Press.

Laniado, D. and Mika, P. (2010). Making Sense of Twitter. In Patel-Schneider, P., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J., Horrocks, I., and Glimm, B., editors, *The Semantic Web – ISWC 2010*, volume 6496 of *Lecture Notes in Computer Science*, pages 470–485. Springer Berlin / Heidelberg.

Lin, T., Mausam, and Etzioni, O. (2012). No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903, Jeju Island, Korea. Association for Computational Linguistics.

Lösch, U. and Müller, D. (2011). Mapping Microblog Posts to Encyclopedia Articles. *Corpus*, 2013.

Mausam, Schmitz, M., Soderland, S., Bart, R., and Etzioni, O. (2012). Open Language Learning for Information Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Meij, E., Weerkamp, W., and de Rijke, M. (2012). Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12*, page 563, New York, New York, USA. ACM Press.

Mendes, P. N., Passant, A., Kapanipathi, P., and Sheth, A. P. (2010). Linked Open Social Signals. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 224–231. IEEE.

Mihalcea, R. and Csomai, A. (2007). Wikify! Linking Documents to Encyclopedic Knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 233, New York, New York, USA. ACM Press.

Millard, D. E., Gibbins, N. M., Michaelides, D. T., and Weal, M. J. (2005). Mind the semantic gap. In *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia - HYPERTEXT '05*, page 54, New York, New York, USA. ACM Press.

Milne, D. and Witten, I. H. (2008a). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence*.

Milne, D. and Witten, I. H. (2008b). Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, page 509, New York, New York, USA. ACM Press.

Nakashole, N. and Weikum, G. (2012). Real-time Population of Knowledge Bases: Opportunities and Challenges. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 41–45, Montr{é}al, Canada. Association for Computational Linguistics.

Narr, S., De Luca, E. W., and Albayrak, S. (2011). Extracting semantic annotations from twitter. In *Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval - ESAIR '11*, page 15, New York, New York, USA. ACM Press.

Ngomo, A.-c. N. and Heino, N. (2011). Federated Knowledge Extraction for Semantic Web Applications. *ACL*.

Passant, A., Bojars, U., Breslin, J. G., Hastrup, T., Stankovic, M., and Laublet, P. (2010). An Overview of SMOB 2: Open, Semantic and Distributed Microblogging. In Cohen, W. W. and Gosling, S., editors, *ICWSM*. The AAAI Press.

Phuvipadawat, S. and Murata, T. (2010). Breaking News Detection and Tracking in Twitter. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 120–123. IEEE.

Presutti, V., Draicchio, F., and Gangemi, A. (2012). Knowledge extraction based on Discourse Representation Theory and linguistic frames. In *EKAW 2012*.

Ritter, A., Clark, S., and Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study.

Rizzo, G., Troncy, R., Hellmann, S., and Bruemmer, M. (2012). {NERD} meets {NIF}: {L}ifting {NLP} extraction results to the linked data cloud. In *{LDOW}, 5th {W}orkshop on {L}inked {D}ata on the {W}eb, {A}pril 16, 2012, {L}yon, {F}rance*, {L}yon, {FRANCE}.

Rusu, D., Fortuna, B., and Mladenic, D. (2011). Automatically Annotating Text with Linked Open Data. In Bizer, C., Heath, T., Berners-Lee, T., and Hausenblas, M., editors, *4th Linked Data on the Web Workshop (LDOW 2011), 20th World Wide Web Conference (WWW 2011).*, Hyderabad, India.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 851, New York, New York, USA. ACM Press.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., and Sperling, J. (2009). TwitterStand. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, page 42, New York, New York, USA. ACM Press.

Shinavier, J. (2010). Real-time #SemanticWeb in <= 140 chars. *IN PROCEEDINGS OF LINKED DATA ON THE WEB 2010, ON WWW2010*.

Spitkovsky, V. I. and Chang, A. X. (2012). A Cross-Lingual Dictionary for English Wikipedia Concepts. In *LREC 2012 Istanbul*.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, page 841, New York, New York, USA. ACM Press.

Thompson, A.-J. and Kemp, E. A. (2009). Web 2.0: extending the framework for heuristic evaluation. In *Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction - CHINZ '09*, pages 29–36, New York, New York, USA. ACM Press.

Vrandečić, D., Ratnakar, V., Krötzsch, M., and Gil, Y. (2011). Shortipedia aggregating and curating Semantic Web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3):334–338.

Wu, F. and Weld, D. S. (2010). Open information extraction using Wikipedia. In *48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics.

Xu, T. and Oard, D. W. (2011). Wikipedia-based topic clustering for microblogs. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–10.

Zhao, W., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing Twitter and Traditional Media Using Topic Models. In Clough, P., Foley, C., Gurrin, C., Jones, G., Kraaij, W., Lee, H., and Mudoch, V., editors,

Advances in Information Retrieval, volume 6611 of *Lecture Notes in Computer Science*, pages 338–349. Springer Berlin / Heidelberg.