Title
Entity Extraction and Disambiguation in Short Text Using Wikipedia and Semantic User Profiles

Permalink
https://escholarship.org/uc/item/85x56504

Author
Zendejas, Ignacio

Publication Date
2014

Peer reviewed|Thesis/dissertation
Entity Extraction and Disambiguation in Short Text Using Wikipedia and Semantic User Profiles

A thesis submitted in partial satisfaction of the requirements for the degree Master of Science in Computer Science

by

Ignacio Zendejas

2014
ABSTRACT OF THE THESIS

Entity Extraction and Disambiguation in Short Text Using Wikipedia
and Semantic User Profiles

by

Ignacio Zendejas

Master of Science in Computer Science
University of California, Los Angeles, 2014
Professor Alfonso F. Cardenas, Chair

We focus on entity extraction and disambiguation in short text communications, which have experienced some advances in the last decade, but to this day remain very challenging. Much of the research that has helped advance the field has leveraged crowd-sourced, external knowledge bases like Wikipedia to build probabilistic and machine learning models for entity extraction. That work has its basis in Wikify! and has recently been applied to understanding the topics discussed on social media where a terse, lossy form of communication makes topic detection even more challenging. We expand on this work and show that on the Twitter data experiments we conducted that leveraging a rich, semantic history of entities that users discuss can improve the accuracy of semantically annotating their future social media posts.
The thesis of Ignacio Zendejas is approved.

Wei Wang

Junghoo Cho

Alfonso F. Cardenas, Committee Chair

University of California, Los Angeles

2014
To my amazing parents, Maria L. and Felipe Zendejas, for always being there for me.

To Mario, Veronica, Felipe Jr., and Alex for all they taught me.

To the love of my life, Heidy, and Thiago Ignacio whom we love so much already.

And to Dr. Alfonso Cardenas for his unwavering support and patience.
# Table of Contents

Abstract

1. Introduction 1

2. Background 4
   2.1 Wikipedia-Based Semantic Annotations 7
      2.1.1 Wikipedia Articles as Entities 8
      2.1.2 Wikipedia Anchor Text as Keyphrases 9
   2.2. Semantically Annotating Social Media Text 12
      2.2.1 Social Man and Social Media 12
      2.2.2 Anatomy of a Tweet 14

3. Related Work 17
   3.1 Wikipedia Mining 17
   3.2 Semantically Annotating Short Texts 18
   3.3 User Histories as Relevant Context 20

4. Hypothesis and Contributions 20

5. System Overview 22
   5.1 System Architecture 22
   5.2 Wikipedia Processing 23
   5.3 In-Memory Storage of Keyphrases, Entities and Profiles 26
   5.4 Two-Step Machine Learning Method For Semantic Annotations 27
      5.4.1 Classifying Keyphrases as Entities 27
      5.4.1.1 Pre-Processing Tweets 27
      5.4.1.2 Entity Classification Features 29
      5.4.2 The Entity Classifier 32
      5.4.3 Ranking Entities 33
      5.4.4 User Profile Representations 35
   5.5 Analyzing Hashtags 35

6. Experiments and evaluations 36
   6.1 Sampling of Twitter Users 36
   6.2 Training and Evaluation Data 39
   6.3 Evaluation Method 42
   6.4 Experimental results 44
      6.4.1 Entity Detection Evaluation 44
      6.4.2 Entity Disambiguation Evaluation 49
      6.4.3 Per User Evalution 54

7. Conclusion 56

9. Appendix A – Manually Labeled Data 58

10. References 58
1. INTRODUCTION

Over the last decade, there has been an increase in usage of social networking sites like Facebook and Twitter. Facebook's founder, Mark Zuckerberg, has posited that the amount of data users generate grows exponentially, with each individual sharing twice as much each year as they did the previous year\(^1\). This presents an opportunity to understand what's happening on a global scale as well as more accurately modeling the tastes and interests of social networking users. While ensuring privacy is fully respected, analyzing what users share both publicly and with friends can help build what is called the interest graph. Such interest graph can be described as directed graph between users and their relationships (the social graph), along with directed edges from these users to millions of entities, their interests.

These entities can represent anything from music bands to public officials and restaurants to movies, or those very broad terms themselves—in effect, just about any person, organization, place or concept conceived by humankind. They, in turn, can also be represented as a graph with edges between two entities indicating some form of semantic relationship between them. Such relationships can involve the more traditional lexical semantics such as synonymy and homonymy but can be best thought of as more conceptual or topical in nature. For example, knowing that Facebook and Twitter both provide a social networking service, can be represented as both Facebook and Twitter having is-a relationships to “Social networking service”, a concept that is a “Web service”, and so forth.

The interest graph has itself sparked a lot of interest as it promises to deliver many benefits to both social media users as well as advertisers, retailers and service providers vying for the former's business or attention. By building an accurate profile of users' interests retailers and content providers, for example, can tailor product or content recommendations, respectively, to

\(^1\) http://techcrunch.com/2011/07/06/mark-zuckerberg-explains-his-law-of-social-sharing-video
their users based on their interests [17]. Facebook, for example, has stated publicly that they use a representation of a user interests to offer a more relevant view of their news feed to avoid overwhelming their users who may have many Facebook friends who over-share\(^2\). They also base their business model in great part by enabling advertisers to specifically target users much more precisely than the traditional demographics-based approach by compiling explicitly stated interests in Facebook profiles (in the traditional user-inputted sense) or implicitly by their browsing behavior on their site and other interactions like clicking the now ubiquitous “Like” button\(^3\).

In order to build these profiles, it has become imperative to analyze the unstructured and often-unlabeled textual content users produce and consume. Automatically understanding human language has proven to be a computationally expensive [1,4,6] and highly inaccurate endeavor [26] in the field of Natural Language Processing (NLP). A decrease in the cost of computing resources has only recently made it possible to train statistical language models at web scale [4] to translate text in one of the few successful NLP stories to date, but even those are no where near solving the Turing test when it comes to human language translation.

Understanding the semantics—the meaning and topics discussed—of text is similarly challenging if not more challenging when it comes to shorter pieces of text that are commonly shared on social networks—especially on Twitter, where users can “tweet” no more than 140 characters at a time. Given these character limitations and the informality of conversations on social networks, users tend to use abbreviations, colloquialisms, and often do not correct for spelling or grammatical errors making it increasingly difficult for any NLP algorithm to infer the meaning of such texts.

\(^2\) https://www.facebook.com/marismith/posts/10151571592800009
Further compounding this difficulty, individuals normally make the safe assumption that their audience has sufficient context to infer the meaning of a post without needing further references or explanations. For example, a student of the University of California, Los Angeles (UCLA) watching a football game will tend to express her support with a message as succinct as “go bruins!” with very little context, knowing full well that other college friends are either watching the same event, know about said event, or are acquainted with the author of the message well enough to know that she's a fan of UCLA's football team. For any automated algorithm, however, “Bruins” can be one of many entities including, among others, members of the educational institution founded in 1919 and located in Westwood, the Boston Bruins hockey team, UCLA's football team or UCLA's basketball team.

The focus of our work is the first, to the best of our knowledge, that builds an end-to-end system which detects and disambiguates any concept, not just named entities, that maps to Wikipedia and leverages a profile of users' interests also mapped to Wikipedia entities to improve both the detection and disambiguation in a feedback-loop fashion. Other work has previously built only subcomponents of our system and usually assumes detection of entities (or the much simpler task of named entities) has been completed, when this in fact is the biggest challenge [13,26] and without this critical step, it makes any real-world usage impractical. The other end-to-end system previously built limits the space of concepts it detects and disambiguates to named entities within six core categories and without using any profile features. We show that modeling user interests by extracting entities that are mapped to Wikipedia articles, such as “UCLA Bruins football,” using a machine learning can help augment future posts with additional context useful in both detecting and disambiguating entities in such posts, thereby leading to even more accurate user profiles. To validate our hypothesis, we used Twitter
data sampled from thousands of Twitter users to both build user profiles and evaluate the precision and recall of annotating Twitter posts, or “tweets”, with Wikipedia topics.

The organization of this thesis follows. The next section will provide more background knowledge necessary to understand our approach and the scope of the problem we're addressing. Section 3 will describe the previous work we build upon. Section 4 will detail our hypothesis and contributions beyond the existing work. Section 5 will present a thorough explanation of how we model users' interests and how we then use such models to improve the accuracy of entity extraction in often short social posts. Given that this approach is meant to be used by real-life systems, section 6 will overview the architecture used to implement our entity extraction algorithm and propose ways of scaling it to millions of users. Finally, section 7 will discuss the results of experiments we used to validate our hypothesis and propose new ways in which our work could be expanded and how it can be applied to other problem domains.

2. BACKGROUND

The idea for this research was inspired by the primary author's own experience working with entity extraction and disambiguation systems in the industry. Despite the many advances he has witnessed annotating long-form text such as web pages to even shorter texts including social media posts or semi-structured profile topics, it was disappointing how such semantically advanced annotators would constantly annotate “bruins” incorrectly to the more highly-linked, probably most correct “Boston Bruins” rather than the “UCLA Bruins”. It's immediately obvious that humans have a vastly superior advantage over computers because they're beneficiaries of a great deal of knowledge about the world and about the events leading up to a piece of communication. It would be advantageous to be able to train computing systems to be
able to learn some of that contextual background information to be able to accurately make sense of the text being shared on social media for reasons we've previously highlighted.

This problem spans other domains like web search and as we will describe in the next section, has been improved successfully with personalized search. As concrete evidence, performing a search for “bruins” on Google's search engine without being signed in with a Google account yields links to the official website of the Boston Bruins hockey team in the top-ranked position. In general, pages referencing the hockey team continue to dominate the top ten results. However, when signed in, Google is able use the primary user's search history to personalize his search results and successfully ranks the UCLA Bruins official site more highly, but still ranks the Boston Bruins official website and other pages related to the hockey team predominantly. In fact, Google still presents the Boston Bruins as the entity in their Knowledge Graph that its algorithms feel is the most relevant at the top-right of the results page as can be seen in the figure below perhaps optimizing for referential use cases rather than interests.

---

4  http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html
Figure 1 – Google's search results for “bruins” when (a) the primary author is not signed in versus when he is signed in and thus, results are personalized based on his history. Note that Google disambiguates “bruins” to the Boston Bruins in all but one case.

Unlike search engines which have the possibility of returning a diverse set of results in
the form of a list and can often leverage user search histories and browsing behavior, entity
extension and disambiguation systems are usually limited to picking the one entity they infer as
the most relevant for any given piece of text. While overall and on average, such system make
the best choice when annotating text for most users, for some users this could lead to inaccurate
representation of their interests and consequently a degraded user experience.

To annotate short text typical of social media, we leverage existing methods which use
Wikipedia as a knowledge base and its articles as references which help disambiguate such
annotations. We describe the background of the work in the following subsection and proceed to
describe the nature of social media posts, specifically those of Twitter, which we use as the basis
for our experiments.

2.1 Wikipedia-Based Semantic Annotations

Wikipedia has withstood the test of time in spite of all the initial lack of trust it garnered
due to the frequency of inaccurate information. It has become a very prominent source of
knowledge for the billions of users who have access to the World Wide Web. Since its inception
in 2001, this encyclopedia has grown to host nearly 4.5 million English articles and expanded to
various other languages\(^5\). The amount of information present in the English version of
Wikipedia comprehensively covers the most notable people, places, organizations and human-
conceived notions including world leaders, popular travel destinations, fortune 500 companies
and recently conceived (relatively speaking) human notions like machine learning. Its breadth
and depth of knowledge also covers specific or less popular topics, albeit less comprehensively
at times, like gradient boosting, which was used extensively in our experiments, and the
Brazilian Telenovela *Sua Vida Me Pertence*, which first aired in black and white television back

in 1951.

All in all, the richness of knowledge which is semi-structured in nature has inspired a great deal of research to both automatically curate Wikipedia automatically or to build probabilistic and machine learning models that are then used to solve complex NLP problems amongst others. Medelyan et al [20] wrote a comprehensive overview in 2009 that is still relevant today and highlights some of the seminal work in the space we expand upon and will overview in the sections below.

2.1.1 Wikipedia Articles as Entities

Semantic annotation of text, a subtask of topic classification which leverages Wikipedia as a knowledge base, like our work, treats Wikipedia articles as referential entities or topics that unambiguously describe a phrase in human language. That is, while traditional NLP focused on parsing text, chunking it into phrases, and identifying a noun phrase's type (person, place, organization, phone number, etc), semantic based annotation will provide richer information that not only indicates a noun phrase represents a person, for example, but it will reference that specific individual by linking the phrase to her corresponding Wikipedia article. Knowing that Clinton is a person can be useful under various applications, but to advance Artificial Intelligence system's understanding of text and expanding on that knowledge, it is useful to know which specific person is meant by a phrase and facts about that person. If the person is William Jefferson "Bill" Clinton (referred to subsequently by his commonly known name Bill Clinton), then a system could model a representation that he is a male like many other prominent figures and was the 42\textsuperscript{nd} president of the United States of America, a country located in the continent of North America. It would also be useful to identify him as the spouse of another prominent person who can also be associated with the phrase “Clinton” (or “clinton”—no capitalization—
on more informal social media text) and so on and so forth.

All those facts can be gathered from the more structured components of a Wikipedia article, namely from infoboxes\(^6\) as well as the more unstructured body of text. Infoboxes are usually located on the top-right section of Wikipedia articles and are based on specifically designed templates for various types of entities or concepts. Clicking on the “View source” link at the top right of Bill Clinton's Wikipedia page shows the article was annotated with the “president” infobox template\(^7\) which lists his date of birth, predecessors, spouse, among other useful facts. The article for the United States, for its part, shows it was annotated with the “country” infobox template which includes facts like the estimated population, capital and even geo coordinates. This infobox, however, does not structurally list the continent the aforementioned country belongs to. However, the first paragraph indicates that its 48 contiguous states are located in the continent of North America, whose phrase links to the corresponding continent's Wikipedia article.

It is this seemingly unstructured content within Wikipedia which we will now focus on as it provides the basis for building our system. It should be noted, however, as we'll detail later that infoboxes were also successfully used to identify certain types of topics like songs and television show episodes which led to inaccurate annotations.

2.1.2 Wikipedia Anchor Text as Keyphrases

Mihalcea and Csomai were amongst the first, if not the first, to realize that anchor texts like “North America” described above could be used as reliable means of identifying candidate phrases or keywords for annotation [22]. Their insight was to take the surface form of linked
phrases on Wikipedia articles like traditional search engines have done for years to identify the unambiguous form, or the Wikipedia target of those links. For example, the surface form “My Life” in Bill Clinton's article links to the unambiguous article titled “My Life (Bill Clinton autobiography)” that describes the former president's autobiography written in 2004. The surface form “my life”, which we convert to its lower case form for the purposes of generalization and to be consistent with how our system pre-processed them, also links to an album by Mary J. Blige, a song by notable rapper 50 Cent, and the “My Life” disambiguation page, which also lists and links to some of the afore mentioned articles amongst several others. Clearly, beyond the fact that the phrase “my life” could refer to the life of any one of the billions of living humans who use social media or have written any form of text, it is also very ambiguous within the more notable selection of entities or concepts within Wikipedia.

Nonetheless, the surface forms of anchors within wikipedia provide a useful signal in determining what constitute prominent noun phrases, or what we call keyphrases based upon Mihalcea and Csomai's notion ofKEYPHRASENESS(p), which is the probability that a phrase, p, was linked on Wikipedia. We'll formally define these concepts in subsequent sections, but it's worth introducing the concept now as we'll use it heavily in the vocabulary of this thesis.

This notion of keyphrases turns out to be a very useful feature in classifying entities and disambiguating them thanks in part to Wikipedia's guidelines. These guidelines are included below for convenience.

---

8 The table referenced uses material from the Wikipedia article “Manual_of_Style/Linking”, which is released under the Creative Commons Attribution-Share-Alike License 3.0. See: http://creativecommons.org/licenses/by-sa/3.0
### What generally should be linked

An article is said to be underlinked if words are not linked that are needed to aid understanding of the article. In general, links should be created to:

- relevant connections to the subject of another article that will help readers understand the article more fully (see the example below). This can include people, events, and topics that already have an article or that clearly deserve one, so long as the link is relevant to the article in question.
- articles with relevant information, for example: "see Fourier series for relevant background".
- articles explaining words of technical terms, jargon or slang expressions/phrases—but you could also provide a concise definition instead of or in addition to a link. If there is no appropriate Wikipedia article, an interwiki link to Wiktionary could be used.
- proper names that are likely to be unfamiliar to readers.

### What generally should not be linked

An overlinked article contains an excessive number of links, making it difficult to identify links likely to aid the reader's understanding significantly. In particular, unless they are particularly relevant to the topic of the article, the following are not usually linked:

- everyday words understood by most readers in context;
- the names of major geographic features and locations; languages; religions; common occupations; and pre- and post-nominals;
- common units of measurement, e.g. relating to area, length, temperature, time, or volume (if both non-metric and metric equivalents are provided, as in 18 °C (64 °F), usually neither unit need be linked because almost all readers will understand at least one or the other unit);
- dates...

| Table 1. Wikipedia's guidelines for linking in articles. |

From the set of guidelines described in the table above, we highlight their emphasis on linking words which are relevant to the article and a call for avoiding linking well-known words or concepts known to readers. For example, there is an article for the word “She”, but it rarely if ever gets linked unless it's in the context of listing or describing other personal pronouns as within the article “He”. The other reason these guidelines are helpful for our purposes is that, as we'll show later, using the outlinks of an article (the edges in the Wikipedia graph) is useful in when identifying relevant and ambiguous entities in a text.

While these guidelines are generally followed, we learned that this was not often the case leading to problems which we had to address and that are detailed in sections 5 and 6. Moreover, the third-person style and other best practices of encyclopedic texts that Wikipedia adheres to
generated several challenges in applying techniques like Wikify!, which was initially conceived to annotate longer pieces of text, to succinct pieces of text typical of social media, which we describe next.

### 2.2 Semantically Annotating Social Media Text

Our work by no means is the first to annotate Twitter with semantics using Wikipedia concepts. We'll highlight some of that work in later sections, but in this section we focus on understanding the nature of text on social media and overview some of the challenges of our task.

#### 2.2.1 Social Human and Social Media

Based on anthropological and scientific evidence overviewed by Christiansen and various multi-disciplinary experts [5], it is traditionally accepted that language has proven to be an evolutionarily beneficial construct developed by humans. It has first and foremost allowed us to communicate more efficiently to survive what must have been more dangerous circumstances in our early history. In its oral form, it went on to strengthen our bonds as social animals as we gathered round the fire for meals and shared stories. These social gatherings also involved lots of wisdom and knowledge that was passed on for many generations. Acquisition and transfer of knowledge was streamlined further with the advent of written communication.

Written language for its part has now made it possible to store vasts amount of knowledge acquired over millennia. The scale of the World Wide Web and even Wikipedia are now testament to the volume of knowledge we now have access to within split seconds thanks to the power of computers and highly advanced software technology.

That scale, however, pales in comparison with the amounts of data that is shared on
social media sites like Facebook, Twitter, Twitter, Pinterest, and others. Facebook has over one billion active users and it alone stores over 180 Petabytes of data each year⁹. A vast amount of that data is probably accounted for by photos, but Facebook also touts lots of written forms of communication, sharing of links as well as more implicit behavioral data that requires an understanding of the text to make better sense of it.

Twitter and Instagram for their part boast over 240 million and 200 million users, respectively¹⁰,¹¹. Twitter gained popularity very rapidly as a microblogging platform and started with people communicating their day-to-day activities and has also evolved over the years as a news-sharing and photo sharing platform. Instagram started as a mobile application for sharing photos with stylistic filters but is in some ways also used as a means to communicate with friends.

These services while often controversial or easily misunderstood have become widely popular because as we've discussed above, humans have a core need to communicate, socialize and share vast amounts of knowledge. The ubiquity of these services, then, opens up the possibility to understand what people are communicating about at a global scale and inferring knowledge from those pieces of communication in the form of useful facts about newsworthy events and even a better understanding of social media users and their interests. At the individual level, being able to find communities of individuals with shared interests has been a natural force for the social man. Search technology offered by these services as well as others, opens up the possibility of discovering such communities of users that have evolved over the years in the web from its earliest forms such as Usenet newsgroups to message boards and now

---

⁹ Note that we extrapolate over 180 PB from > 0.5PB/day times 365 days a year. [https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920](https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920)
¹⁰ [https://about.twitter.com/company](https://about.twitter.com/company)
¹¹ [http://blog.instagram.com/post/80721172292/200m](http://blog.instagram.com/post/80721172292/200m)
social networking sites.

On a global scale, having a semantic and automated way of analyzing vast amounts of communications on these platforms now makes it possible to gather intelligence about newsworthy events, potential for more relevant forms of advertising, and a means for improving personalized experiences across the web. To understand what's trending and newsworthy, many news organizations have turned mostly to Twitter as most forms of communication are popular and in many cases involve the news makers, or prominent world leaders or celebrities which many people care to learn more about.

Marketing departments have, in turn, begun to leverage these services to reach more global audiences than traditional forms of media like television and radio. The challenge has been, however, that they have less visibility about the content their advertisements are being displayed with and can only trust that companies like Facebook and Twitter are able target their ads to the right audience. Our work gets us a step closer to solving these challenges as it tries to understand the text shared by users and in turn build more accurate models of users' interests, which are then used to better understand future texts by that user more effectively.

We based our experiments on publicly accessible Twitter data and thus we will focus on describing such communication platform next.

2.2.2 Anatomy of a Tweet

Twitter is a micro-blogging site that gives anyone with an email the ability to communicate their whereabouts, their present activities, and their interests publicly and in real-time. What makes Twitter popular is that openness and its appeal to a diverse set of users and audiences that span the gamut of interests: from health enthusiasts to people who just like to
share photos and celebrities who communicate with their fans to journalists and their organizations who are hoping to quickly catch the next newsworthy story. The public nature of data on Twitter also makes it a source of research in computer science as well as other fields.

Media shared on Twitter are commonly referred to as “tweets”. Each tweet is limited to no more than 140 characters and can include a variety of special tokens that are used to augment a tweet with meta data or references to other entities, which are not to be confused with semantic entities as we've described above. Such tokens can include mentions of other twitter accounts by prefacing the account's Twitter user name with an at “@” sign. Users can also include hashtags that are topical annotations for a given tweet and are prefaced with a pound sign “#” and link to a search page on Twitter of other tweets containing that phrase or tag. Finally, tweets can also contain URL entities normally shortened by services such as Bitly or Twitter and which redirect users from the bit.ly or t.co servers, respectively, to the originally shared hyperlinks. These hyperlinks will usually reference news articles, videos, blog posts, images and related media that users found interesting enough to share with their audience, or followers in Twitter nomenclature. Re-tweets (or retweets from now on) are the equivalent of email forwarding in that they propagate tweets by a reader to the latter's own audience. They are usually prefaced by the “RT @<mention>” or postfixed by the “via @<user_name>” formats. Twitter has enabled retweeting more natively, without exposing the afore mentioned templates and instead embeds that data deeper in the metadata, but many users still use the traditional approach of prepending the actual abbreviations.

The limits on how much can be written combined with the fact that the above metadata and URLs count towards those text limits often leads to creative ways in which users often communicate on Twitter by using various abbreviations. An increase in usage on mobile phones
further leads to users further foregoing all grammatical and capitalization standards for convenience and expediency's sake, much like in text messages, which is where the 140 character limit comes from (Twitter was initially relied on Short Message Services SMS on mobile phones to relay communications, so they imposed a 140 character limit, leaving 20 bytes of data for meta data, including twitter user names)\textsuperscript{12}.

All of these conventions that have now expanded beyond Twitter onto Facebook, Instagram and other related media sites and vice versa, result in a nightmare scenario for individuals trying to make sense of this lossy, unstructured data with machine learning and traditional forms of NLP. The consequence is that traditional NLP algorithms don't do very well at parsing text on social media [26], so new ways to analyze this text are needed and rely on a huge deal of domain expertise to build the models or heuristics, really, to handle the various ways in which people abbreviate. To give an example of the challenges posed to traditional NLP algorithms, see the tweet below:

\begin{center}
\textbf{If u ate a bagel everytime a VC said "KAUFFMAN" here at #VIC2012, u'd weigh 222 pounds like me}
\end{center}

\textbf{Figure 2: An example of a terse, lossy tweet.}

The above is an example of a tweet which our systems uses for evaluating our work. The use of the letter u to replace the word “you” is an example of Internet slang, commonly known as netspeak or more formally as translexical phonological abbreviations. Also, “everytime” is considered an alternate form of the phrase “every time”, but for an automated system which uses Wikipedia to label tweets, “everytime” can also be a keyphrase for a song by pop star Britney

\textsuperscript{12} http://www.140characters.com/2009/01/30/how-twitter-was-born/
Spears, whose name is commonly misspelled. For the curious or those trying to replicate our experiments, we include a link to the full set of labeled data we used to validate our hypothesis in the Appendix section.

3. RELATED WORK

Our system is an end-to-end system and therefore touches many sub-areas of research in natural language processing (NLP). The primary area our work helps advance is that of using Wikipedia as a knowledge base to annotate text with rich semantics. Secondly, our system is tailored to short text with a special emphasis on social data as it is a new landscape ripe with opportunities due to the volume of data. And, finally, it uses a user’s history to improve the accuracy of future annotations, thereby, also creating more accurate profiles which can be used for personalization, among other applications. We begin with the latter space, as it's only been recently investigated.

3.1 Wikipedia Mining

The idea of using Wikipedia as knowledge base to expand the semantics and understanding of text for various applications including text classification and semantic annotation, go back to the seminal work of, but not limited to, Gabrilovich et al [12] and Mihalcea et al's Wikify! [22], respectively. Medelyan et al [20] do a comprehensive job overviewing this space and serves a great introduction into the benefits of leveraging Wikipedia for NLP and machine learning tasks due to its vast amounts of crowd-sourced information rich with semantics and semi-structured data. Our work is most similar to Wikify! as it was one of the first systems proposed to leverage anchor data in Wikipedia to process text more accurately;
however, like most early work, Mihalcea et focused on long-form text documents that mostly assumed proper grammar and capitalization. Also relevant within the early work is that of Cucerzan et al [7] and Milne et al [23], as they pushed the area further with improved methods for linking and disambiguating large texts using a notion of semantic relatedness now popularly referred to as commonness.

3.2 Semantically Annotating Short Texts

With the rise of social networks, came the realization that traditional methods of entity extraction and disambiguation were unsuitable for short, lossy text typical of sites like Twitter. Ferragina et all [10] were the first to focus on short texts with the highly cited TAGME system, followed by the work of Meij et al [20]. The latter demonstrated improvements over the TAGME system as it specialized on semantically annotating tweets and introduced novel features and methods of handling twitter data.

This latter work treats the problem of annotating tweets as a search retrieval problem whereby effectively the contents of a tweet are used to identify the most relevant entities or Wikipedia pages. They use a learning to rank approach with a machine learning model and also demonstrate how poorly traditional systems trained on larger pieces of text do on shorter text. Though our work uses similar machine learning models and even features, ours treats the problem differently and is much more suitable for higher precision tasks, whereas Meij's work is more suitable for higher recall tasks. They are the only group that made their evaluation data available, but we found that it was unsuitable for our purposes for various reasons; first, the problem designs were different, and we could not treat the whole end-to-end system as a search and rank problem; second, as they acknowledge in their Error Analysis section, their annotators
had several false positives and false negatives and we were able to corroborate their findings as we found that our system was more exhaustive than their annotators making it further difficult for us to evaluate our system and get an apples-to-apples comparison.

Other work around annotating, or linking as some of the authors call it, entities to Wikipedia have mostly focused on the second component of our system, which is that of disambiguation [8, 27,28]. As Ritter et al [26] and more recently, Guo et al [13] have demonstrated (and we further demonstrate), the entity detection task is the most challenging on short text, specially on social data, making it impractical to use their systems alone. Ritter and his co-authors show that out-of-the-box named entity recognition (NER) systems obtain a max f-score of about 44. After further work by them, they were able to boost the f-score to about 67.

In some ways, the work that most resembles ours is that by Guo et al [13] as they, too, build an end-to-end system which does both detection and disambiguation. Unbeknownst to us until after our system was built, we used several similar features, but unsurprisingly since both our systems were influenced by other work in the space. The difference between their system and ours is that we 1) use profile-based features, and therefore, 2) would not benefit from the limited set of categories as they do entity detection and disambiguation for six core categories only (Person, Location, Organization, TV Show, Book/Magazine, Movie), making their system more in line with a traditional NER. Again, we do not limit the space of concepts because both our goal and requirements are to build a very accurate profile of user interests, and limiting it to a few categories would reduce our chances of doing that because there are a slew of topics that people are interested in beyond those that have proper names and/or titles, including but not limited to types of dishes or cuisines, music artists and genres, academic interests, and more.

For a summary of the area of research involved with making sense of social data, there
are two recent papers which do a more comprehensive job see [3] and [15].

3.3 User Histories as Relevant Context

The use of profiles built atop a history of an individual's activity traces its roots in that of personalized search. That space is very well established and some of the earlier work can be traced back to several highly cited articles by Speretta et al [31], Sontag [30], and Quiu and Cho [25], and Liu et al [18], with Dou [9] providing a survey early in that space. Our work further establishes like such earlier work in that space, that when little to no context is available about the intent of a user, it's often useful to leverage a user's history of implicit and explicit interests evidenced in past activity.

Shen et al [28] were amongst the first, if not the first, to model user interests using Wikipedia to then aid in the disambiguation task. However, as previously noted above, they've assumed the entity mentions have been previously detected, a task that turns out to be the toughest on social media. Another, limitation of their system is that they also only focus on named entities (a semantic NER approach), reducing the space of entities and therefore richness of profiles to (people, organizations, places and similar types that are more easily identified in traditional text via capitalization features). Other work that has explored the area of building a profile of user interests includes that by Xu et al [32], Michelson et al [21]. However, this work does not, to our knowledge, use such profiles for future topic extraction and/or disambiguation tasks.

4. HYPOTHESIS AND CONTRIBUTIONS

The goal of our work is to improve the accuracy of both entity extraction and
disambiguation of short text that is common on social networking sites and likewise, the accuracy of user modeling of interests as these two models are inter-dependent within our system. We hypothesize that using a history of previously discussed topics extracted from previous posts can improve the recall of entities extracted from future posts by the same author while maintaining a high degree of precision that is required in user-interfacing systems. In short, we wish to expand the contextual knowledge that a system trying to annotate short texts has about the author to be able to more accurately infer the interests and the meaning of the authored posts.

Our primary contribution is a flexible machine learning method that processes a piece of social media text incorporating the additional knowledge of its author's interests to determine the best semantic annotations for that text. It consists of a two-step approach that first treats the identification of keyphrases that represent relevant entities as a binary classification task and upon positively classifying a keyphrase as an entity treats its disambiguation as a learning to rank problem using another machine learning model. To the best of our knowledge, ours is the first that builds an end-to-end system for both detection and disambiguation of entities—and not just named entities like people, places or organizations, but any concept notable enough to have encyclopedic reference.

As a secondary contribution that is in line with our goals, we propose a simple yet novel approach to extract and disambiguate entities from hashtags, which pose the added difficulty often representing compound phrases concatenated without spaces. We show that this leads to an improvement in the overall precision and recall of semantic annotation of the text in which the hashtags are contained.
5. SYSTEM OVERVIEW

In this section, we summarize the system we built in order to process social media data to extract and disambiguate relevant entities mentioned in texts that map to Wikipedia articles and build a model of users interests using those entities. First, we give a high-level overview of the system as a whole. As an important requirement when using Wikipedia, we follow that overview with a discussion of the ways in which we processed Wikipedia and the types of models we built with its encyclopedic knowledge. Next, we describe how we use Wikipedia entities to represent user interests. Thereafter, we detail the two-step approach we use to perform the semantic annotation task and the discriminative machine learning models we used discussing their benefits and disadvantages. Finally, we present the simple, yet novel approach of handling the difficult-to-read hashtags by automated systems.

5.1 System Architecture

Our system consists of four major subsystems. The first is the Wikipedia processing engine which parses the raw Wikipedia dumps to generate the data models necessary to annotate text. The second is an in-memory system that stores all this data, as well as the user profiles. Third, we built a pre-processing engine which takes a tweet and generates two sets of features which get fed into the machine learning engine and stores the results in the database. The last one is the machine learning engine which takes as input a raw vector representation of the features, classifies a keyphrase as an entity and then ranks the most relevant one with a second machine learning model. We detail each next.
Figure 3 – A diagram of our end-to-end system. First, we pre-process a tweet producing groups of keyphrases that are converted to feature vectors with help from an entity and keyphrase database derived from Wikipedia. Next, if a keyphrase is classified as an entity, its referenced entities get transformed to feature vectors and scored. Finally, the highest ranked entity for each keyphrase group is added to the user profile, which gets used in both steps.

5.2 Wikipedia Processing

In order to build the data models necessary for entity extraction and disambiguation, we first had to process Wikipedia. We built a simple yet easily parallelizable framework using Apache Spark\textsuperscript{13} to take the large XML-based dump of Wikipedia provided by the Wikipedia

\footnote{\textsuperscript{13} Apache Spark Official Site. http://spark.apache.org/}
foundation in July, 2013\(^{14}\). Next we extract the anchor texts within the Wikipedia articles, build the probabilistic models for keyphraseness within Wikipedia and the commonness (i.e., the probability that an entity disambiguates a phrase, given that it was linked by that anchor text).

We also extracted other useful data which was useful within the disambiguation step, such as the out-links from each article, the infobox types they were annotated with, and the Wikipedia categories which they referenced.

Like other approaches which inspired our work, we filter all the internal Wikipedia articles which discuss or represent the various templates, categories, users, images, files and more. We also filtered out all articles representing lists like “List of Internet phenomena”, index pages and disambiguation pages. To account for lack of proper capitalization and punctuation of text within social media, we first pre-processed a Wikipedia article's text eliminating all punctuation except for apostrophes in possessive keywords like “McDonald's” and converted all text to its lower case form. The same was done for anchor text, resulting in very clean text.

In computing the keyphraseness of a phrase, we first identified the cases within articles in which a phrase appeared as an anchor and for the remaining text, we processed it using that very dictionary set of anchors to identify all articles in which anchors were not linked and therefore contributed negatively to their keyphraseness. Then to compute the keyphraseness of the set of surface forms which appeared as anchors, we filtered out all instances which appeared only once throughout the Wikipedia corpus. And after computing the keyphraseness value (which the reader will recall is the total number of documents in which a phrase was linked over the total number of documents in which it appeared either linked or unlinked), we filtered instances which had values lower than 0.01. We also filtered out keyphrases which were composites of a set of stop words (e.g., “the”, “out there”, etc). Had we needed to process grammatically correct

\(^{14}\) Wikimedia Downloads. http://dumps.wikimedia.org/
English, we would have kept cases that were composites of stop words as they may represent names of songs, as in the case of “OutThere”, among other titles of entities. However, these would introduce a lot more problems when processing the very lossy and noisy text found on social media.

To compute the commonness of an entity given a keyphrase we first resolved all targets including redirects to a their canonical form (i.e., the title of the final article destination of a link or redirect). A redirect in Wikipedia represents an intermediate page which simply redirects the user to a final destination. Redirects are commonly used for well-established aliases of prominent entities (e.g., “Obama” and “Barack Hussein Obama” redirect to the “Barack Obama” article), to compensate for commons misspellings (e.g., “Britany Spears” redirects to “Britney Spears”), or to handle changes in names to Wikipedia articles [e.g., in cases where there are collisions on names, some entities that are less common may change a disambiguating phrase that is normally found in the title of an article within parenthesis such as in “ABC News (Australia)”]. Since redirects also provide a good source of keyphrases by handling misspellings which are also bound to occur on social media as in the Britney Spears case, we normalize the surface form of redirects and if there are no collisions with existing keyphrases, we add it as a keyphrase with a conservative keyphraseness of 0.25.

After resolving all targets by following any redirects, we joined all these targets, t, by their keyphrase, k, (after a filtering step to avoid bloating memory unnecessarily), summed their frequencies of each t for that a given k and normalized by the total number of targets to compute the commonness measure which is effectively a prior probability that a given phrase will disambiguate a phrase. Moreover, for each article (or entity in our case), we extracted a list of all resolved targets (i.e., other entities) which it linked to. We also extracted a list of Wikipedia
categories like “Australian news television series” that are linked to by each article at the bottom.

Finally, we also processed the infoboxes and looked for such cases that represented titles of songs or musical pieces as well as specific television episodes as they were causing problems because they were represent very common expressions like “out there”, but without stop words, that rarely occurred within Wikipedia in unlinked form because the case sensitivity and more formal third-person language of Wikipedia.

5.3 In-Memory Storage of Keyphrases, Entities and Profiles

The output of this pre-processing stage was then loaded into a MongoDB\textsuperscript{15} instance with 12GB of memory running on a desktop computer with an Intel i7 processor to ensure that all necessary data fit in memory. Given the small nature of our test set, we only touched a small section of Wikipedia so we only required about two GB of memory space. The data was learned into two collections that each stored the set of keyphrases and entities respectively. To avoid expensive join operations for the large amounts of data, each of those collections stored all instances of keyphrases and entities in one big unnormalized document. However, any join operations between these two collections was indeed handled by business logic. For production-based systems, we would recommend using a faster, truly in-memory key-value store like Redis\textsuperscript{16} since this dataset is generally read-only and it's easy to keep backups and replicate for improved load balancing. For our purposes, MongoDB proved a reliable alternative to SQL-based data stores because it did a good job of caching data in-memory and that made rapid experimentation possible despite the size of the records.

To store user profile data, we used a database to initially store and capture the Twitter

\textsuperscript{15} MongoDB Official Site. https://www.mongodb.org/
\textsuperscript{16} Redis Official Site. http://redis.io/
data. However, after sampling the users' tweets which we evaluated on, we chose to instead quickly serialize the data in JSON format and write it to disk. Paying a small I/O cost upfront meant that we could later query the in-memory hash table much more quickly to retrieve and update user data. In a real-world application, this implementation would not be ideal as it does not provide any persistence guarantees.

5.4 Two-Step Machine Learning Method For Semantic Annotations

As discussed above, our method for extracting entities and disambiguating them involves a two-step approach and two machine learning models for each step. The first step takes a social media post, pre-processes its text and generates a vector representation of a set of features which then get classified as an entity or non-entity. For any keyphrases classified as entities, our system then performs a second step and takes a subset of those same features plus an additional set and inputs a matrix of the feature vectors for each candidate entity and scores each vector returning the highest for each keyphrase-to-entities group. For the purposes of experimentation, however, the processes was slightly different and was prototyped as a set of offline Scala and Python scripts which ran processes for the different steps. For the purposes of this thesis, we omit these implementation details and instead focus on the high-level design and algorithms used.

5.4.1 Classifying Keyphrases as Entities

5.4.1.1 Pre-Processing Tweets

To determine if a phrase is a relevant entity within a social media post, we first preprocess the text, stripping all punctuation except for possessive cases of the apostrophe to
match the normalized form of the Wikipedia data. In the case of tweets, we do further pre-process further by performing the following filtering and transformations:

- First, we strip all “retweet” metadata such as “rt @<user_name>” and “via @<user_name>” template forms described in the background section of this thesis. While the primary author of this post (i.e., the one whose tweet is getting retweeted) may be relevant to the meaning of the post itself, we found that this was not always the case and that it could lead to inaccuracies in building user profiles.
- Next, we strip addressees of a tweet for the same reason we outlined above.
- Third, we take mentions which do not prefix the tweet as in a reply and therefore, probably refer to a relevant entity and resolve the user names to the full name of the user using the Twitter API. Because some of the names of Twitter users were very common names, but they themselves were not notable enough to have a Wikipedia article, this could often lead to inaccurate disambiguation of their names. Therefore, we took the extra step of filtering out any twitter mentions of users whose Twitter account was not verified and had at least ten thousand followers – a good measure of notability.
- Finally, we transform hashtags from their machine unfriendly form and try to deduce the actual unigrams when they're composites of words. We'll describe this approach in detail in section 5.4.

After performing the steps above, we use sliding windows of n-grams, where n is equal to one and up to six words, and perform in-memory database lookups to retrieve all matching phrases within our keyphrase dictionary. The set of keyphrase are then grouped by overlapping cases to avoid detecting multiple entities per phrase and we generate a set of features as will be
described below, on a per group and per keyphrase basis.

5.4.1.2 Entity Classification Features

After employing the steps above, we processed the step to generate a vector form representation of a tweet that was then feed into a discriminative trained regressor or classification. We'll detail the regression models used below, but first we'll have to describe the feature set we used. For the purposes of experimentation, we used regressors and not binary-style classifiers because the former produced scores which give more flexibility in determining at what threshold to operate and thus, yield nice Precision vs. Recall curves that make it easier to evaluate different models and feature sets. Likewise, they make it possible to build more accurate models by operating at higher thresholds and thus optimizing for a high enough recall while guaranteeing a level of precision that does not degrade the quality of the user profiles.

The regression problem thus involved taking a set of features and trying to predict a value between zero and one with lower scores interpreted as likely not being an entity and values closer to one showing a stronger signal for the classification of a positive case.

The table below details the set of features we used. Some of the features have been used before and as expected the Wikipedia-based keyphraseness itself had to be one of these features as it is core to the approach we employ in this first step. The novel features are ones that involve a similarity, or semantic relatedness between a keyphrase's entities and the user's profile, which we'll discuss further in a subsequent sub-section.
## Wikipedia-Based Entity Classification Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Formula</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEYPHRASENESS(p)</strong></td>
<td>( K(p) = \frac{</td>
<td>D_a</td>
</tr>
<tr>
<td><strong>MAX-CMNS(x)</strong></td>
<td>( \max(CMNS(e,p)) ), for all phrases in a group as determined by their overlapping regions within a tweet, and where ( CMNS(e,p) ) represents the number of times ( e ) was linked when ( p ) occurred as an anchor in Wikipedia.</td>
<td>Purpose: Higher values represent less ambiguity. As shown by Meij et al [20], commonness is a good signal and usually helps easily identify notable named entities or concepts.</td>
</tr>
<tr>
<td><strong>NUM-TARGETS(p)</strong></td>
<td>The total number of targets of a keyphrase, ( p ), within Wikipedia.</td>
<td>Purpose: Higher values means more ambiguity. When little context is present, we prefer our system err on the side of false negatives than false positives.</td>
</tr>
<tr>
<td><strong>CMNS-ENTROPY(p)</strong></td>
<td>(-1 \times \sum (p(CMNS(e)) \times \log(CMNS(e)))).</td>
<td>Purpose: A uniform distribution of CMNS values of a keyphrase, ( k ), represents higher levels of ambiguity. When little context is present, we prefer our system err on the side of false negatives than false positives.</td>
</tr>
</tbody>
</table>
| **RELATEDNESS(p,E)**     | \( \text{REL}(p,E) \) is the sum of relatedness of some phrase, \( p \), versus that of the top 10 most relevant entities referenced by other contextual keyphrases in a tweet. Here, we define relatedness as: \[
\text{rel}(x,y) = \log(\max(|E_x|,|E_y|)) - \log(|E_x \cap E_y|) / (\log(|W|) - \log(\max(|E_x|,|E_y|))),
\] where \( E_x \) are the outlinks and inlinks of an entity \( x \), and likewise for \( y \), \( E_y \) represents the entities that link and are linked from \( y \). Note: this differs somewhat from other implementations in the literature. |
### Purpose:
This represents a measure of overall coherence of a phrase versus its surrounding context.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SUM-CMNS-BADT(p)</strong></td>
<td>The sum of the commonness (CMNS) of entities pointed to by p, which represented songs and television episodes. This was done to ameliorate the case where common phrases in everyday speech are used as titles of songs or other relevant media. Purpose: As discussed in the Keyphraseness feature, this was meant to reduce the chance that we'd pick up every day phrases like “no excuses” as a song because those phrases aren't very common in Wikipedia unless they refer to a notable song or television episode. We leave the exploration of using a similar feature to down-weight titles of movies and other media.</td>
</tr>
<tr>
<td><strong>MIN-LEV(p)</strong></td>
<td>The min of the Levenshtein [16] distance between p and the titles of the Wikipedia entities it links to. Purpose: Very broad terms like “food” get a low keyphrases, so this is meant to ameliorate the problem and is also meant to prevent the system from picking up people from common first or last names, which are not really referenced in a tweet.</td>
</tr>
<tr>
<td><strong>AVG-PROFILE-RELATEDNESS(p)</strong></td>
<td>The average semantic relatedness of the entities linked to by p against the user's profile, where relatedness was computed as detailed above. Purpose: Meant to provide additional context, especially when little to none exists within a tweet.</td>
</tr>
</tbody>
</table>

### Tweet-Based Entity Classification Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NUM-SUPERSTRING(p)</strong></td>
<td>Number of keyphrases in a tweet, t, which are superstrings of the keyphrase p. Purpose: We did this to boost longer phrases like “cream cheese” over “cheese”, regardless of their keyphraseness.</td>
</tr>
<tr>
<td><strong>CAPS-LEFT(p)</strong></td>
<td>The number of capitalized words left of the keyphrase, x, in a tweet, t, normalized by the total number of words that are capitalized in t. Purpose: Not a very successful feature to ameliorate the problem with picking up proper names from substrings of a person or named entity that does not have a Wikipedia article. For example, our system picked up the state of Virginia from a...</td>
</tr>
</tbody>
</table>
person's first name.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPS-LEFT(p)</td>
<td>The number of capitalized words right of the keyphrase, x, in a tweet, t, normalized by the total number of words that are capitalized in t.</td>
<td>Purpose: See explanation for CAPS-LEFT.</td>
</tr>
<tr>
<td>PREPS-LEFT(p)</td>
<td>The number of prepositional phrases left of the keyphrase, x, in a tweet, t, normalized by the total number of words that are prepositions.</td>
<td>Purpose: Again, intended to capture cases like above.</td>
</tr>
<tr>
<td>PREPS-RIGHT(p)</td>
<td>The number of prepositional phrases right of the keyphrase, x, in a tweet t, normalized by the total number of words that are prepositions.</td>
<td>Purpose: Again, intended to capture cases like above.</td>
</tr>
<tr>
<td>NUM-KEYPHRASES(t)</td>
<td>The number of phrases in a tweet, t, that map to Wikipedia keyphrases.</td>
<td>Purpose: This feature was meant to help our system potentially down-weight the influence of the user's profile if sufficient context was present.</td>
</tr>
<tr>
<td>NUMBER-TOKENS(t)</td>
<td>After pre-processing, the number of unigrams or words in a tweet, t.</td>
<td>Purpose: Same as above. We believe that for shorter text, coherence with a user profile is important.</td>
</tr>
<tr>
<td>LEN(p)</td>
<td>The number of alphanumeric characters in the keyphrase p. Like NUM-SUPERSTRINGS meant to bias for longer phrases.</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 – A description of entity classification features we explored.

Note that we used a combination of features which used Wikipedia based signals at the keyphrase level as well as signals from the post itself.

5.4.2 The Entity Classifier

In order to learn a model which accurately classified a keyphrase as a relevant entity within a tweet, we used several state-of-the-art regression algorithms provided by the Python-
based scikit-learn package [24]. We choose regression algorithms as they offer the flexibility of determining the best threshold depending on the application. For the purposes of our application, it is important to build highly accurate user profiles as those get fed back as features for later regression tasks. Therefore, we would operate at a high enough threshold that guaranteed at a precision of at least 0.85. At the disambiguation phase we operated at a lower threshold that was close to a value of zero to maximize recall given that the disambiguation rankers were very accurate (with an accuracy of over 0.85) regardless of the threshold.

In our experiments, we found that SupportVectorRegressors (SVR) [2] and GradientBoostedRegressors (GBR) [11] performed the best during the entity classification phase. For the disambiguation phase, given that there were nearly 15,000 instances of entities to score, SVR was very slow to train, so we opted to use GBR instead which still took about one to two minutes per one-user-out validation set. Meanwhile, validating entity classification took under a minute per user (hold out set).

5.4.3 Ranking Entities

For all keyphrases that are classified as relevant entities in a social post, the second step, again, involves disambiguating that phrase by rank-ordering the entities that keyphrase points to within Wikipedia using a number of features which we list below. As with keyphrase classification, we treat this problem as one of regression to predict values between 0 and 1. After feeding a matrix-based representation of all entities linked to by a keyphrase, we chose the highest scored entity as the final output unless the disambiguation score was lower than threshold, $T$. The reasoning meant that even if the first step scored the keyphrase as an entity, it was so ambiguous that none of the entities scored highly and so any system trying to achieve a
high precision should take the conservative approach and also threshold the disambiguation scores.

The set of features we used follows next.

<table>
<thead>
<tr>
<th>Disambiguation Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMONNESS(k,e)</td>
</tr>
<tr>
<td>Given a keyphrase which links to e, the commonness CMNS score of e.</td>
</tr>
<tr>
<td>Purpose: The prior probability on Wikipedia that a phrase refers to an unambiguous entity is a strong indicator.</td>
</tr>
<tr>
<td>RELATEDNESS(e,p)</td>
</tr>
<tr>
<td>The semantic relatedness as computed by as described in the table above for an candidate entity, e, against all other entities referenced by keyphrases in the tweet.</td>
</tr>
<tr>
<td>Purpose: Measures the coherence of entity to its surrounding context.</td>
</tr>
<tr>
<td>IS-BAD-TITLE(e)</td>
</tr>
<tr>
<td>A value of 1 or 0, in the case where an entity represents the title of a song or TV episode or not, respectively.</td>
</tr>
<tr>
<td>Purpose: We completely eliminated common phrases that represent song titles or specific episodes, but kept them for potential context (as in cases where an artist and song are mentioned together).</td>
</tr>
<tr>
<td>LEV-DIST(p,e)</td>
</tr>
<tr>
<td>The Levenshtein distance of a keyphrase p, and the entity being scored.</td>
</tr>
<tr>
<td>Purpose: The intent was to learn whether it was more correct to link more general instances of a topic (e.g., a specific version of a game versus a series) automatically. Labelers predominantly favored more specific cases when available.</td>
</tr>
<tr>
<td>RANK(e,k)</td>
</tr>
<tr>
<td>The rank of an entity e, given that k is a keyphrase.</td>
</tr>
<tr>
<td>NUM-INLINKS(e)</td>
</tr>
<tr>
<td>The total number of articles that link to e within Wikipedia.</td>
</tr>
<tr>
<td>Purpose: A cheap way to approximate PageRank to understand which entities are more influential.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile-Based Disambiguation Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>INLINKS-FRM-PROF(e,p)</td>
</tr>
<tr>
<td>The total number of entities in the user profile that link to e.</td>
</tr>
</tbody>
</table>
| Purpose: A measure of coherence against a user's interests, where we assume that users will usually discuss or share media related to their interests which are relatively well represented by the
As can be seen in the table above, some of the features carried over form the first step but were computed for each candidate entity only as opposed to overall entities linked by a keyphrase extracted in a tweet.

### 5.4.4 User Profile Representations

In order to classify social media posts more accurately using our proposed approach of leveraging contextual knowledge about an author's interests, we first bootstrapped a user profile. For the purposes of our experiments, we benefited from first gathering a substantial amount of posts for each user we evaluated against. For live systems, one can use any of the methods discussed in the related sections first to build an accurate profile. Another alternative is to use our two-step approach without introducing the profile features, operating at reasonably high thresholds.

### 5.5 Analyzing Hashtags

In order to improve the precision and recall of entity extraction and disambiguation of tweets, we found it necessary to expand hashtags as they often provided very salient and prominent entities but were effectively undetectable without first splitting them. That is, we had
the challenge of determining how best to split a hashtag like “#hmvDemiLovato” or even worse "#singularityu”. To solve this problem we cleverly realized that many of the hashtags had a camel-cased representation as in the first case described above. For a human reader, the best way to split the first hashtag may eventually jump out as doing so whenever the letter casing changes. That is, "hmvDemiLovato” should be come “hmv Demi Lovato” and after doing a search, not knowing the context, one could verify that indeed that was the best way to split the hashtag as the artist Demi Lovato made an appearance at an HMV store in the United Kingdom. But how does one solve the problem of “singularityu” which has no upper cased characters to provide any hints on splitting? We devised a simple approach that queries any hashtag using Twitter's search API and scan the results of the query for other variations of a hashtag which do present the camel case. We then take the most common case of a camel-cased form and split it as described for the first case above. Thus, for “singularityu” our system found that “SingularityU” was the most popular camel-cased form and split the hashtag correctly to “singularity u”, which unlike “singularityu”, was a keyphrase that pointed to the right entity, “Singularity University”.

6. EXPERIMENTS AND RESULTS

This section presents the experiments that were carried out to validate our hypothesis. The first sub-section overviews the approach taken to gather evaluation data by sampling a set of Twitter users. Next, we'll describe how a ground-truth set was generated. Finally, we'll describe how well our system was able to use profile features and how it compared to several baselines.

6.1 Sampling of Twitter Users

Because our system relies on rich profiles of social networking users as determined by
prior entities they discussed, we randomly sampled 20 Twitter users to evaluate our entity extraction and disambiguation approach. Twitter proved to be a useful source for evaluation data not only because it is our goal to make sense of what is discussed on networking sites like Twitter, but because users publicly post or “tweet” their day-to-day activities, news articles or other online media they found interesting, their whereabouts and more. Thus, by gathering data on a set of users, we didn't risk invading anyone's privacy as a result of our experiments.

For the purposes of evaluating our approach we sampled 20 individuals from a set of about one thousand Twitter users for which we collected tweets over the course of about 6 months and which had more than 100 authored tweets or retweets. Although we could have randomly sampled users, we chose to take a more measured approach when sampling users. Our goal was to sample for diversity across a different set of criteria. We chose a good distribution of users according to how prolific they are as measured by the number of total tweets they posted during said period. The average number of posts per user was approximately 318, with a standard deviation of about 870. As with lots of social data, the distribution of the average number of posts adhered to a power law. The maximum number of posts was over 32000 by an online media organization that, quite representatively of similar organizations, tweeted links to articles authored by its writers.

We also sampled fairly evenly according to the topicality of users' posts. We measured topicality by the number of entities per tweet that were extracted using our approach, but without introducing profile features. Likewise, we considered the number of keyphrases per tweet. This was done to ensure that we didn't bias for users that either tended to not discuss many topical interests or the other extreme which favors our system, since it would lead to more contextual data about a user's interests. User topicality was represented by a more gaussian-like
distribution, with a few outliers. Bar graphs of both entities per post and keyphrases per post are found below.

![Histogram of entities (a) and keyphrases (b) per post for all users before sampling.](image)

Figure 4 - Histogram of entities (a) and keyphrases (b) per post for all users before sampling.
To further account for the diversity of Twitter users, we took into account the dominant category of a user's posts to avoid biasing towards categories or clusters of topics that favor our entity extraction approach. For the purposes of reproducibility, one can manually check the identity of a twitter account and its description or inspect a few posts by a user to get a sense for what the topical preferences or tendencies are. For example, some accounts are by their Twitter description and/or organizational affiliation tailored to specific topical interests such as movies, technology news, and so forth. However, we didn't actively target specific accounts to include in our dataset and rather took a more quantitative approach before accepting or discarding a twitter account which we describe next.

To sample the 20 users, we took the various statistics described above and grouped users across quantiles in each set of these distributions. We then randomly sampled from each quantile without replacement while still trying to sample evenly across all quantiles. Each of these users was then further inspected to, as described above, account for diversity of topics, to exclude users whose tweets were predominantly in a non-english language, and users that mostly shared links with little accompanying text.

6.2 Training and Evaluation Data

After 20 users were successfully picked to account for as close as possible to an unbiased distribution across the various metrics discussed above, used the most recent 25 posts we had in our database for each user as the holdout, or evaluation set. This gave us a total of 500 posts on which to evaluate. The remaining posts, which we call the user history for short, up to those most recent 25 tweets were used to build user profiles by running our system without profile features. We thereby built a user model of their interests as described in section 5.4.4 using the
entities extracted in those posts and stored them in a JSON-based\textsuperscript{17} blob as described in Section 5.4.4 for quick loading into memory. We'll show that even having a more imprecise algorithm to generate this prior contextual set of interests is useful in improving the results of future extraction. When building user profiles, it should be noted that only entities extracted from the tweet itself were used. While we could have used entities extracted from the web pages linked by the shortened URLs for improvements in overall accuracy, we chose not to in order to get a better evaluation of our algorithm as it is meant to be robust in cases when little context is provided—in even the shortest of posts such as “Let's go bruins!” as previously discussed.

In order to accurately evaluate the performance of our method, as well as alternatives, it was critical to get a clean and highly accurate dataset. Therefore, we developed a web application to allow us to more quickly and accurately label tweets with the most accurate entities. The web interface included a section that displayed the text of each user's tweet. Below it, a list of automatically extracted keyphrases from the tweet's text was also displayed with the ability to click on them for a convenient way to quickly retrieve the disambiguated entity that was predicted by the labeler to best represent the entity referenced by such keyphrase. Given the encyclopedic nature of Wikipedia keyphrases, and the various heuristics to filter phrases deemed unlikely to represent a keyphrases as discussed in section 5.2, it was important to not only label a tweet with entities that were reachable via the automatically extracted keyphrases. Otherwise, the evaluation set would be too favorable to Wikipedia-based entity extraction methods like ours that rely heavily on keyphrases. Therefore, a search field with auto-completion was provided for both keyphrase and entity lookups, so that labelers could quickly add other relevant entities.

The ability to search keyphrases and tweets not explicitly mentioned in the tweets allowed labeling of highly ambiguous phrases and abbreviations like “lps” which, upon

\textsuperscript{17} ECMA-404 The JSON Data Interchange Standard. http://www.json.org/
close investigation by a human and in consideration of the user's context, referred to limited partners in a business venture, primarily in the venture capital sense and therefore the most relevant entity “Limited partnerships” was used. Unfortunately, no keyphrase-based entity extraction approach is able to extract the best representative entity “Limited partnership” given that there is no prominent keyphrase “lp” which can be analyzed to arrive at such an entity. While this no doubt affected the results of our evaluation we, again, wanted the most accurate evaluation possible and did not omit such entities from the evaluation set. (There are ways to handle these cases, but we didn't in our case due to time constraints, but they wouldn't be too difficult to incorporate within our framework.) Other examples of this rigorous approach to labeling included labeling twitter mentions even if the user name was some transformation or abbreviation of a known Wikipedia entity. For example, there were several mentions of “@danariely” that were disambiguated to Dan Ariely, the well-known behavioral economist, who administers a Twitter account under the aforementioned username. Exceptions of mentions not disambiguated to Wikipedia entities included instances of retweets or relays from one user to another's followers via the classic “RT @<original_author>” user-initiated protocol; or when a mention was indicative of a reply or conversation as determined by the standard Twitter protocol of usually pre-pending tweets with a user's name with an optional dot to broadcast the reply to others beyond those who follow all parties involved in the conversation (e.g., “.@danariely I like your book!”). Such mentions were stripped entirely per the former reasoning in the previous section using a simple regular expression. When kept, the “@” was stripped allowing for the keyphrase extractor to match mentions against any found in the database or was looked up using the Twitter API to obtain the user's full name as described in the previous section.

Likewise, hashtags were stripped of the hash character “#” and looked up or resolved as
described in the previous section. Like mentions, they tend to be abbreviations or concatenated phrases thereby making them difficult to disambiguate without applying some specialized resolution techniques like the simple and novel approach we proposed. And as stated above, the contents of a hyperlink were not used to train or evaluate any of the approaches. Therefore, all hyperlinks detected by a regular expression open-sourced by Twitter\textsuperscript{18} were stripped to avoid picking up entities such as Youtube, from very popular un-shortened urls.

The end result of this meticulous manual labeling of 500 tweets was that there were a total of 1,216 entities that directly map to Wikipedia articles. Of those, 173 Wikipedia entities were unreachable by keyphrase-based approaches, but were nonetheless not removed to avoid biasing the results of evaluation. To view the evaluation set, please see Appendix A.

6.3 Evaluation Method

To evaluate the performance of our unambiguous entity extraction method and to compare it against other methods which do not rely on user profiles, we first took the micro-averaged precision and recall scores, since user-profile agnostic approaches would treat each tweet independent of any context and without regard to the author.

\[
P_{\text{micro}} = \frac{\sum_{i=1}^{\left|C\right|} TP_i}{\sum_{i=1}^{\left|C\right|} TP_i + FP_i}; \quad R_{\text{micro}} = \frac{\sum_{i=1}^{\left|C\right|} TP_i}{\sum_{i=1}^{\left|C\right|} TP_i + FN_i}
\]

The micro-averaged precision, \(P_{\text{micro}}\), is the sum of all true positives, \(TP\), for all users, \(C\), over the sum of all true positives and false positives, \(FP\), for all users. Likewise, the micro-averaged recall, \(R_{\text{micro}}\), is the sum of all true positives divided by the sum of all true positives and

\footnote{\textsuperscript{18} https://github.com/twitter/twitter-text-java}
false negatives, FN, for all users. In our case, the total recall will be taken as a percentage of the total number of manually labeled entities (1,216) across all 500 tweets while the precision will vary depending on the number of true positives and false positives that were retrieved. To further summarize the micro-averaged precision and recall, we also provide an $F_1$ score, which is defined as follows:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$ 

To perform the actual evaluation, we used a one-user-out approach. That is, we trained on the manually labeled tweets of 19 users and evaluated on the remaining user's tweets, repeating this step 20 times for each user. This achieved two goals: 1) by training on the manually labeled set, we ensured that our machine learning models were unaffected by noise in more automated labeling systems, and 2) by not cross-validating across random samples, as is usually done, we reduced the chances that our models would overfit for specific users.

As part of this overall evaluation process, a user's 25 tweets were processed in chronological order. In the case of our two-step approach, after any entities were extracted, they were added immediately to the user's profile. This was meant to mimic a real-life system in which users' tweets need to be processed in real-time. For all other approaches, each individual tweet could be processed in any order.

To further understand how well our system performs, we show results for each user separately. To provide additional context, we plotted their individual results as a function of several statistics, like the number of posts in their user history, the number of entities in their profiles, the number of entities per post that were automatically extracted, and the number of entities in the manual dataset both overall, and on average per tweet.
6.4 Experimental Results

Soon after trying the first implementation of our approach, we learned that there was a downside to relying on the full set of entities in a user's profile. The recall improved over the baseline approach significantly but the precision suffered lowering the overall F₁ score. After debugging we learned that it was because certain keyphrases were disambiguated incorrectly because if an entity in the user's profile was referenced by such keyphrases, but with a very low commonness they would obtain a high enough relatedness to overcome the former's feature score. For example, in one of the posts “state” would incorrectly be disambiguated to “United States” because the country had previously been added to the user's profile, affecting the profile relatedness scores (note that the similarity between the same entity will always be 1). Thus, for every candidate entity, we only computed relatedness measures against all other entities in a user's profile except itself. In practice, this is preferable since false positives could continue to reinforce themselves and never give other entities in the future with richer textual context a chance to rank higher. As such, a side-effect of this step is that it minimizes inaccuracies in a user's profile and keeps the feedback loop less noisy.

6.4.1 Entity Detection Evaluation

Because our approach performs end-to-end entity detection and disambiguation, it was important to optimize its performance in correctly detecting entities. That involved classifying keyphrases as entities or non-entities. We show the results of our one-user-out validation in the figure below and follow it up with a table summary and a discussion. We note that some of the ground truth entities in our labeled set were unreachable from the set of all keyphrases our system extracted, so the actual recall we display below does not represent the overall entity detection recall. However, any false negatives will be accounted for in the next step when we
evaluate the overall performance of our system against the ground truth labels.

![Figure 5](image)

Figure 5 – Precision Recall of Entity Classification. (SVR = support vector regressor)

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>Max F₁ Score</th>
<th>Rec. @ 0.9 Prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyphraseness</td>
<td>0.8153 (+0.0%)</td>
<td>0.7599 (+0.0%)</td>
<td>0.3774 (+0.0%)</td>
</tr>
<tr>
<td>SVR</td>
<td>0.8816 (+8.13%)</td>
<td>0.8155 (+7.32%)</td>
<td>0.6257 (+65.8%)</td>
</tr>
<tr>
<td>SVR + Profile</td>
<td><strong>0.8967 (+9.98%)</strong></td>
<td><strong>0.8244 (+8.48%)</strong></td>
<td><strong>0.7013 (+85.8%)</strong></td>
</tr>
</tbody>
</table>

Table 4 – Summary of entity classification results

In entity recognition alone, the use of the Wikipedia-based keyphraseness didn't do so poorly, but it clearly is not suitable for social media text as its recall suffers at high enough precision values. The graph and table show that adding profile features boosts both precision and recall especially at higher precision values. At 90% precision (the recommended lower-bound we recommend for live systems to avoid adding noise to user profiles), for example, the difference in recall (not accounting for unreachable cases) is nearly twice as high as
keyphraseness alone (about 85.8% higher) and nearly 8 points higher than without using the profile feature alone (about 10.8% better). These higher recall scores at high precision values are also crucial for the next step where we combine both the entity classification results and the disambiguation scores. For context, it should be noted that even a five percent gain on NLP problems is a reasonable achievement [29]. For a live system processing in the order of hundreds of millions of social media posts per day\(^\text{19}\) that represents millions of more accurately detected entities. Our experiments scored three orders of magnitude fewer keyphrases, but was nonetheless still significant at 2,728 total keyphrases). We also note that when false positives occurred, the profile features weren't usually to blame and instead were due to other issues that we've previously discussed in processing terse, lossy text on Twitter that affected all approaches as evidenced that the red (with profile) and blue (without profile) features converged at both ends of the graph.

Using profile features provided additional, useful signaling to our models in cases when little to no textual context was available and it also helped boost some very ambiguous phrases like “bears”, which can represent any species of mammals in the Ursidae family, the Chicago Bears football team, any of the University of California, Berkeley collegiate teams known as California Golden Bears (as was the case in our evaluation set), or any of the other entities listed in Wikipedia's Bear (disambiguation) page\(^\text{20}\). However, these cases were few to show a high lift versus not using profile features alone. As noted before, our system added features like CMNS-ENTROPY to force it to err on the side of avoiding false positives if a keyphrase is too ambiguous and score its entity detection lower if there is not enough contextual information, so in the cases like “bears” it had other scores which boosted its effective confidence enough to

---

19 http://www.washingtonpost.com/business/technology/twitter-turns-7-users-send-over-400-million-tweets-per-day/2013/03/21/2925ef60-9222-11e2-bdea-e32ad90da239_story.html
classify it as an entity at higher thresholds.

To understand the importance of certain features, we used cross-validation and measured the worth of individual features by using an SVM-based approach [14] using Weka [33] which removes one feature at a time. The results of the feature ranking are shown below:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Average Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG-PROFILE-RELATEDNESS</td>
<td>1 +/- 0</td>
</tr>
<tr>
<td>RELATEDNESS</td>
<td>2 +/- 0</td>
</tr>
<tr>
<td>NUM-SUPERSTRINGS</td>
<td>3 +/- 0</td>
</tr>
<tr>
<td>KEYPHRASESENSS</td>
<td>4 +/- 0</td>
</tr>
<tr>
<td>MIN-LEVENSHTEIN</td>
<td>5 +/- 0</td>
</tr>
<tr>
<td>KEYPHRASE-LEN</td>
<td>6 +/- 0</td>
</tr>
<tr>
<td>SUM-CMNS-BADT</td>
<td>7.2 +/- 0.4</td>
</tr>
<tr>
<td>NUMBER-TOKENS</td>
<td>7.8 +/- 0.4</td>
</tr>
<tr>
<td>NUM-KEYPHRASES</td>
<td>9 +/- 0</td>
</tr>
<tr>
<td>CMNS-ENTROPY</td>
<td>10.8 +/- 1.66</td>
</tr>
<tr>
<td>PREPS-RIGHT</td>
<td>10.9 +/- 0.3</td>
</tr>
<tr>
<td>PREPS-LEFT</td>
<td>12 +/- 0</td>
</tr>
<tr>
<td>CAPS-RIGHT</td>
<td>13.1 +/- 0.3</td>
</tr>
<tr>
<td>CAPS-LEFT</td>
<td>14.4 +/- 0.49</td>
</tr>
<tr>
<td>MAX-CMNS</td>
<td>14.4 +/- 2.06</td>
</tr>
<tr>
<td>NUM-TARGETS</td>
<td>15.4 +/- 0.8</td>
</tr>
</tbody>
</table>

Table 5 – Results of 10-fold feature ranking for entity classification using SVM method.

As can be seen in the table above, the profile feature ranks highest, even above keyphraseness, though one should keep in mind that this is only when removing one feature at a time, so what this only shows is that the profile feature is able to combine well with others as was our experience whenever we added seemingly useless features. And it should come as no
surprise that capitalization does not significantly improve performance demonstrating why any traditional entity recognition system wouldn't do so well as was proven by [26]. To get a better sense of why the profile feature is useful, we also show its histogram along with that of the relatedness and keyphraseness features for context.
Figure 6 – a) Histogram of keyphraseness values; b) Histogram of relatedness values; c) Histogram of avg. profile relatedness. (red = entity, cyan = non-entity)
6.4.2 Entity Disambiguation Evaluation

To test the disambiguation step we first grouped all entities by overlapping keyphrases in a post, used one-user-out approach to scoring all entities and then rank-ordered them by their score within each group, independent of whether the phrase was a classified as an entity or not. In this case, we combine the results of the first step with that of disambiguation and evaluate against the manual set, including all unreachable entities from the full set of keyphrases.

![Figure 7 – Precision recall curve of overall results. (GBR = gradient boosted regressor; AVG (PR, R, EP) is the average of profile relatedness, relatedness and commonness)](image)

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>Max F1 Score</th>
<th>Rec. @ 0.9 Prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBR + Profile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG (PR, R, CMNS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAGME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As can be viewed in the graphs and plots above, when combining keyphraseness alone and then ranking with the commonness (CMNS) score, what we consider here as the baseline, the results are much lower than in the entity classification step. Introducing more features as we have within our framework to address some of the challenges in analyzing social media text improves the $F_1$ score at least 10%, but the recall at very high precision more than doubles when using profile features. We also benchmarked the only publicly available system, TAGME [10], which extracts and disambiguated the full set of entities in Wikipedia, not just named entities. As shown in our results, our baseline effectively does better as the aforementioned system tends to pick up a lot of noise in our evaluation set. However, one must keep in mind that such system was not trained on our dataset, but we include it for context.

Overall, introducing profile features lifts the curve much more noticeably when combining both steps in our system. The F-score is 3.7% higher when using profile features versus when they're left out. Meanwhile, the AUC goes up about 7.9% and for production-ready systems, the recall goes up 35% at a solid 0.9 precision value. This value is even higher when at the disambiguation phase we simply take the average of the commonness (CMNS), relatedness and profile relatedness scores, the third model in the table above.

We also note that having introduced the unreachable entities into the evaluation, the max...
recall gets boosted 6% over the baseline and 1.5% when using profile features over the machine learning model without profile features. Finally, we note that all curves, especially the blue curve (representing the profile-less method in our system), have a drop in precision at higher thresholds because the threshold is applied during the entity classification step and even if an entity classifier accurately classifies an entity with a very high score, the disambiguation can rank-order the wrong entity highest leading to false positives. This further shows that using profile features is more robust.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Average Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG-PROF-REL</td>
<td>1 +/- 0</td>
</tr>
<tr>
<td>COMMONNESS</td>
<td>2 +/- 0</td>
</tr>
<tr>
<td>OUTLINKS-TO-PROF</td>
<td>3 +/- 0</td>
</tr>
<tr>
<td>IS-BAD-TITLE</td>
<td>4 +/- 0</td>
</tr>
<tr>
<td>INLINKS-FRM-PROF</td>
<td>5 +/- 0</td>
</tr>
<tr>
<td>LEV-DIST</td>
<td>6.3 +/- 0.46</td>
</tr>
<tr>
<td>NUM-INLINKS</td>
<td>6.7 +/- 0.46</td>
</tr>
<tr>
<td>RANK</td>
<td>8 +/- 0</td>
</tr>
<tr>
<td>RELATEDNESS</td>
<td>9 +/- 0</td>
</tr>
</tbody>
</table>

Table 7 – Results of 10-fold Feature ranking for disambiguation using SVM method.

We show the rank-ordered features and summarize the results in the table above. A profile feature used when ranking at the disambiguation phase was consistently the highest ranked feature by the SVM method used in [14]. The other profile-related features also fared well in the third and fifth positions.
Figure 8 – a) Histogram of commonness values; b) Histogram of relatedness values; c) Histogram of avg. profile relatedness. (red = entity, cyan = non-entity)
6.4.3 Per User Evaluation

To understand how our system performs on an individual user by user basis, we compared precision, recall and f-cores across different statistics at a reasonable threshold (one that could be used in production):

- the total number of entities in the profile prior to evaluation
- the average number of entities per tweet prior to evaluation
- the total number of entities that were manually labeled per user
- the average number of entities per post in the manual set of tweets

The only insight worth noting here but that would require further investigation with more users is that as one would expect with the overall results above, as the size of the entities in the profile increases, the recall and therefore the F₁ score goes up. However, the precision falters a bit over as compared to users with smaller profiles and even in comparison to the baseline as and to the models which do not use profile features. We leave for further studies, with more users for statistical significance, the task of understanding how time-sensitive user's interests are and the optimal number of entities that should be kept in a user's profile. For these 20 users, it appears the top 500-1000 entities might be optimal, but again, further analysis and more data is needed to determine this value.
Figure 9 – $F_1$ score (a) and precision (b) as a function of the number of profile entities.
7. CONCLUSION AND FUTURE WORK

In this thesis, we have overviewed, to the best of our knowledge, the first end-to-end system which 1) detects or extracts entities, which includes non-named entities, 2) disambiguates such entities by referencing the most relevant Wikipedia article, and 3) manages very ambiguous cases well by leveraging a user profile of the author's interests as additional context. We've done this via a very flexible machine learning approach that, first, treats the entity detection problem as a binary classification problem using a discriminative regressor that leverages user profile features amongst others, and second, after classifying a positive entity, disambiguates to the most relevant Wikipedia article by using another regression model gained trained using user-profile features.

We have further demonstrated that our two step approach using two regression models can do much better than baseline approaches without needing to train on more than 20 users making it practical for use in real-life systems. Adding to the practicality of our approach, we have demonstrated that we can achieve a 50% recall at 90% precision, which given the difficulty of the task, is in our opinion, a considerable achievement.

Finally, we have validated our primary hypothesis that using a simple model of a user's interests and likewise, deriving simple features can lead to more accurate annotation of social data, even in cases where a noun phrase is too ambiguous and/or there is very little textual context surrounding it. As part of this feedback loop, we have shown that this can enable building much more accurate profiles useful for personalization tasks, amongst others.

We leave for future work, the following three very interesting experiments which would be critical for a real-life system. The first is understanding of the time-sensitivity of user's interests and whether a decaying factor is needed to maintain an accurate profile to minimize the
noisiness of such profile and the accuracy of annotation of a user's future posts. Second, it would be valuable to investigate how if, at all, the Wikipedia category graph can be used to get a broader understanding of a user's interests and how that may be used to further increase the recall of annotations, since higher level categories potentially provide a less sparse representation than the entity-to-entity graph of user's interests. We hypothesize that annotating short texts as we've done with Wikipedia articles would make it easier to classify such texts into broader, higher-level categories which are derived from the category graph in Wikipedia. That is, knowing that even a short piece of text as “go bears” could be classified into sports, and with an understanding of the user's interests, more specifically into a category like “college sports”. This task would likely be interdependent with the second and each would benefit from the accuracy of the other. Given the flexibility of our machine learning approach, it would not be difficult to add another set of features to leverage this categorical information, but in our initial investigations we found that the Wikipedia category graph requires extensive pre-processing to address among other issues, the fact that it's represented by a cyclic graph, and would be difficult to extract an accurate, more hierarchical taxonomy.

Lastly, we wonder if our approach could be used to annotate search queries and improve the relevancy of results via personalization. This would pose many challenges, especially in gathering search queries, which tend to be more privacy-sensitive; fortunately, competitions like the 2014 Entity Recognition and Disambiguation (ERD) Challenge 21 has a short text track which includes precisely this kind of dataset.

---

8. APPENDIX A – Manually Labeled Data

Because the size of the labeled data would amount to over 20 pages in this thesis, we're providing a publicly accessible hyperlink below. The data is stored in a TSV format and has the following attributes:

1. UserId – The Twitter id of the user
2. TweetId – The id of the tweet. (Note: that chronological ordering can be obtained without retrieving the tweet by noting that higher values indicate more recent posts.)
3. Text – The actual, unmodified text of the tweet.
4. [Entities] – A variably sized list, including an empty list, of entity names, whose label corresponds to the title of a Wikipedia article. (Note: that some article names may change.)

https://dl.dropboxusercontent.com/u/1703644/zendejas_thesis_labeled_data.tsv (approx. 80KB)

9. REFERENCES


