Entity and Aspect Extraction for Organizing News Comments

ABSTRACT

News websites give their users the opportunity to participate in discussions about published articles, by writing comments. Typically, these comments are unstructured making it hard to understand the flow of user discussions. Thus, there is a need for organizing comments to help users to (1) gain more insights about news topics, and (2) have an easy access to comments that trigger their interests. In this work, we address the above problem by organizing comments around the entities and the aspects they discuss. More specifically, we propose an approach for entity and aspect extraction from user comments through the following contributions. First, we extend traditional Named-Entity Recognition approaches, using coreference resolution and external knowledge bases, to detect more occurrences of entities in comments. Second, we exploit part-of-speech tag, dependency tag, and lexical databases to extract explicit and implicit aspects around discussed entities. Third, we evaluate our entity and aspect extraction approach, on manually annotated data, showing that it highly increases precision and recall compared to baseline approaches.

1. INTRODUCTION

Media platforms, like CNN[1] and Al Jazeera[2], deliver the latest breaking news on various topics about everyday events. Moreover, they provide the possibility to write comments about any published article and engage in discussions with other users. Figure 1 shows an example of a news article about the Scottisch independence referendum, published at Al Jazeera on September 19, 2014. On this article, users started several threads of discussions about *"Scots origin"*, *"Benefit of terrorists from Scottisch independence"*, *"The results of the vote"*, and other subjects. Although user comments form such well-defined discussions, they are displayed in an unstructured way, listed based on time and date information, as shown in Figure 1. Consequently, it is hard for the reader to catch the flow of discussions and to understand their main points of agreement and disagreement. Thus, there is a need for organizing user comments to (1) have a better understanding of the viewpoints related to each topic and (2) facilitate the participation in discussions and thus increase the chance of acquiring new viewpoints. A natural way to summarize and organize comments is to cluster those that contain similar discussions. In other words, they talk about the same entities and argue about the same aspects of those entities. To achieve this task, we need to extract entities and aspects from user comments.

Figure 1: User comments on a news article

The core problem of our work is entity and aspect extraction from unstructured text. Several techniques for Named Entity Recognition (NER) have been proposed in the literature [1, 3, 5–7, 11, 12, 19–21], including supervised, semi-supervised, and unsupervised techniques. Similarly, several approaches have been proposed for aspect extraction from user reviews about products [22]. Exploiting existing approaches to extract entities and aspects from user comments brings new challenges: (1) a user comment can refer to other comments when talking about entities, which requires the

[1] www.cnn.com
[2] www.aljazeera.com

integration of coreference resolution strategies and tailored techniques for identifying the context of a comment; (2) user comments are informal and might include noisy information such as abbreviations and misspellings, which make entity extraction difficult; (3) aspect extraction techniques limit aspects to attributes and components of entities and thus do not cover other forms of aspects related to entities that are not objects, such as people or events.

In this paper, we propose an approach for entity and aspect extraction from user comments on news media platforms, tackling the problems above. The main contributions of this paper are summarized as follows:

1. We extend traditional Named Entity Recognition (NER) approaches to detect more occurrences of entities in a given comment. To achieve that, we use coreference resolution techniques to detect entities from comments that are not self-contained. Furthermore, we exploit external knowledge bases to solve the problem of abbreviations and misspellings.

2. We propose an aspect extraction approach to detect both explicit and implicit aspects around discussed entities. To handle aspect extraction, we exploit part-of-speech tags, dependency tags, and lexical databases.

3. We evaluate our entity and aspect extraction approach, on manually annotated data, showing that it highly increases precision and recall compared to the AIDA, Zemanta, and NERD approaches.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 introduces the problem we are tackling in this work. Section 4 presents the extensions we propose for entity extraction tools. Section 5 describes our approach for extracting explicit and implicit aspects. Section 6 presents and discusses experiment results, and finally Section 7 concludes the paper.

2. RELATED WORK

Our work falls into the category of Text Mining. To the best of our knowledge, we are the first to touch the area of organizing opinionated texts from user-generated content using a domain-independent approach for entity and aspect extraction. One of the most similar works to ours was done by Hu et al. [8, 10]. They extract product features that are commented on by the user, then they group up review sentences commenting on the same features. This work was later improved by Poria et al. [15] who developed an unsupervised technique of product feature extraction and summarization using handcrafted rules. The difference between these approaches and our work is that they completely break down user comments into small sentences and they lose connection to the original user comments, whereas we want to preserve this information. Moreover, they are domain-specific while our work is more general. In terms of beautiful visualization of text summary, the Word Cloud[3] may be one of the best work available. It emphasizes important terms, however, it does not preserve semantic relations between terms. Some other similar works are those focusing on the area of topic mining [9, 14]. Pons-Poratta et al. [14] explored a topic discovery system to browse and select topics of interest. Similarly to topic classification approaches, our work aims at maintaining relationships between entities and aspects. The main difference is that the relationships we build are not only semantic but they depend also on the co-occurences of terms in user comments.

Directly related to our work is research in Named Entity Recognition (NER) [20]. The first class of approaches is supervised and consists in learning classifiers from training data for entity recognition [1, 3, 5, 6, 11, 19]. One of the most well-known open tools for supervised learning of NER is the StanfordNLP[4], which is based on the CRF classifier [7]. To address the difficulty of obtaining manually-annotated data for training, a second class of approaches was proposed. It relies on a semi-supervised learning technique called "bootstraping" [12]. Typically, this approach uses a small initial set of seeds as a start of the learning process, and then it searches for the occurrence of the seed in the sentences and identifies common contextual clues of the initial seeds. Then, the technique proceeds to find new instances of the seeds that have similar context.

The third class of approaches to NER uses unsupervised learning techniques, typically by making use of open knowledge bases such as DBpedia.[5] Some of the unsupervised NER tools are available online, such as AIDA,[6] AlchemyAPI,[7] NERD,[8] Zemanta,[9] and Wikifier.[10] In addition to detecting entities, they are also able to correctly disambiguate entities that have the same name (to some extent). These tools have been evaluated to have good performances [17]. However, their application in our work raises several problems. First, they do not have a common agreement of what an entity is and thus they provide, in some cases, unreliable results. For example, AIDA does not recognize "Tories", the british political party, as an entity, while NERD returns "true freedom" as an entity, which is subjective. This problem makes it hard to decide which NER tool to use for our purpose. Second, they do not cover all forms of occurrences of entities, including coreferences, which is crucial for user comments. In our work, we address these issues to discover more occurrences of entities and reduce the amount of noise involved in the process.

Aspect extraction has also gained attention in the product review domain [22]. These approaches define aspects as the components and the attributes of a given entity. Moreover, they rely on the presence of sentiments to detect aspects. Thus, the process of identifying and extracting aspects and entities is limited to evaluative texts. However, in our work, we want to extend the definition of aspects so that (1) it includes other forms than attributes or components and (2) the extraction does not rely solely on the presence of sentiments.

[3] http://www.wordle.net/

[4] http://nlp.stanford.edu/
[5] http://dbpedia.org/
[6] https://gate.d5.mpi-inf.mpg.de/webaida/
[7] http://www.alchemyapi.com/
[8] http://nerd.eurecom.fr/
[9] http://www.zemanta.com/
[10] http://cogcomp.cs.illinois.edu/page/demo_view/Wikifier

3. PROBLEM DEFINITION

User comments talk about entities and argue about specific aspects of those entities. An *entity* can be either a person, an organization, a location, or any well-defined concept such as languages, nationalities, or wars. By contrast, an *aspect* is all what is arguable about an entity. Consider the following example:

> *"I'm all for Scottish independence. This vote is about continuing to depend on the UK without having to bear the responsibilities."*

This comment talks about the entities *"Scottish"* and *"UK"*, and argues about the aspect *"independence"* of the entity *"Scottish"* and the aspect *"responsibility"* related to the entity *"UK"*. In this work, our task is to extract from each comment the set of entities it talks about, and the aspects related to each entity.

Entity extraction from user comments brings new challenges to traditional NER. First, existing NER tools provide, in some cases, results that do not correspond to our entity definition, such as *"true freedom"*. Note that an entity must refer to an objective concept since subjective concepts are subject to discussion among users and they need to be extracted as aspects. Distinguishing entities from aspects is important for an accurate representation of the various view points of users about a given entity. Second and more importantly, a comment is typically a part of a discussion which implies that entities are not always explicitly mentioned but appear as coreferences. Note that limiting the entity extraction process only to occurrences that mention the name will not allow having a wide coverage of the aspects of an entity. Third, comments are informal in nature and might contain grammatical errors, misspellings, abbreviations, and unreliable capitalization.

To address the above problems, we need first to filter the outcome of NER tools to have only entities of the types we have defined. Second, we need to extract the context of each comment from its related comments and eventually from the news article itself. The underlying structure of comments in news sites is a tree, where users can post a standalone comment, a reply to a standalone comment, or a reply to another reply. Figure 2 shows the two largest sub-trees of the full tree of comments on the Scottish independence news article. The largest contains 28 comments discussing about terrorism, and the other one has 17 comments discussing about Scots origin. Generally, the full tree has a news article as a root, and comments as either intermediate nodes or leaves. Based on this structure, the context of a comment is given by its ancestors. Having context information would then help resolving entities that occur as coreferences and ambiguity related to abbreviations and missspellings.

After extracting entities from comments, we need to extract for each entity its corresponding aspects. Aspects occur in the text as noun phrases. Thus, aspect extraction consists in finding noun phrases that have a grammatical relationship with the extracted entities. Consider the following example:

> *"Scots love to spout nonsense about independence."*

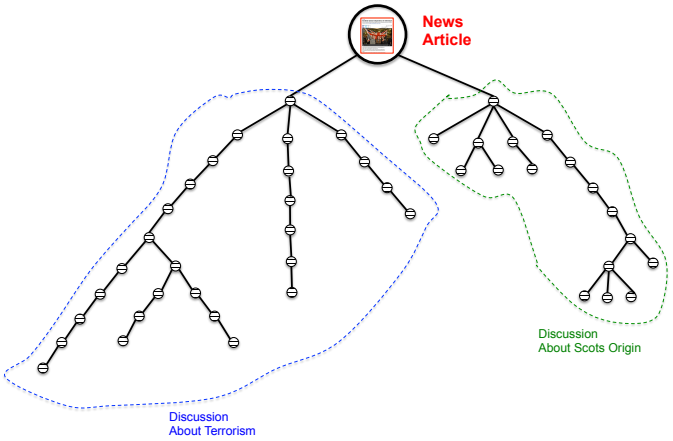


Figure 2: Example of Comment Trees

In this comment, there is a dependency between the noun phrase *"independence"* and the nominal subject of the sentence which is the entity *"Scots"*. Thus, the noun phrase *"independence"* is extracted as aspect for the entity *"Scots"*. The main challenge is that aspects are not always explicitly mentioned in the text. Consider the following example:

> *"You live in the British Isles. The largest is Great Britain. Some of you are Irish, Scottish, and English. Some of you claim to be Welsh."*

The sentence *"The largest is Great Britain"* talks about the aspect *"Isle"* of the entity *"Great Britain"* even though it is not explicitly mentioned. In other cases, aspects can be semi-implicit. Consider the following example:

> *"Scotland can vote however it wants, it's the Scottish peoples right."*

This comment talks about the aspect *"voting"* of the entity *"Scotland"*. In this case, the aspect can be derived from explicitly mentioned words in the sentence such as verbs and adjectives. To tackle the problem of aspect extraction from user comments, we propose in this paper tailored techniques for each case of explicit, implicit, and semi-implicit aspects This involves the exploitation of context information to find frequent co-occurrences of noun phrases with verbs and adjectives. .

4. ENTITY EXTRACTION

Our aim is to exploit an existing NER tool for extracting entities from user comments. However, there are a number of limitations raised by the structure and the nature of user comments as described earlier. In this section we tackle these limitations through the following strategies.

4.1 Entity Filtering

We have defined an entity to be either a person, an organisation, a location, or any well-defined concept such as languages. According to this definition, an entity is an individual of a class. A class is not an entity because it describes a very general concept. The issue with existing tools is that they do not have an agreement on what an entity is. Consider the following example:

> *"I say let the Scots have their true freedom. Don't be afraid of Rasmussen or NATO, this is none of their business. If he ends up sending USS Ronald Reagan aircraft carrier to the coast of Scotland, then he should have done the same to Crimea."*

From this comment, AIDA[21] does not recognize *"Crimea"* as an entity whereas NERD[17] does. Moreover, NERD extracts *"aircraft carrier"* and *"true freedom"* as entities. However, *"aircraft carrier"* is a class and *"true freedom"* is not an instance of a class but an aspect which is arguable.

Our entity filtering approach consists in exploiting a knowledge base to check whether a given noun phrase is an entity or not. To this end, we choose DBpedia [2], a large scale knowledge base extracted from Wikipedia. DBpedia is a graph database that uses the RDF[11] format. It represents Wikipedia categories as resources and uses the *rdf:type* predicate to state whether a resource is a class or an individual of a class. Using this property, entity filtering removes all results produced by NER tools that have no property *rdf:type* in DBpedia.

In the previous example, NERD extracts as entities: *"Scots"*, *"true freedom"*, *"NATO"*, *"USS Ronald Reagan"*, *"aircraft carrier"*, *"Scotland"*, and *"Crimea"*. It provides, for each extracted entity, a link to Wikipedia. We then convert the Wikipedia link to its corresponding DBpedia *URI*, and check whether the object represented in the *URI* has property `rdf:type` other than `owl:class` and `owl:thing`. In this example, because *"true freedom"* and *"aircraft carrier"* do not have any `rdf:type` property, they are removed. All other entities have types other than `owl:class` and `owl:thing` and thus they are kept.

4.2 Name Normalization

When writing comments, users tend not to write the full name of entities. They use abbreviations, or only last names of people, and sometimes they misspell names. We call all variations of the name of an entity an *"alias"*. Interestingly, traditional NER tools are not always able to recognize entities from aliases. Thus, we introduce a name normalization technique that converts all aliases to normalized names to facilitate entity extraction. To begin, we extract entities from the news article and all its related comments using the entity filtering technique described earlier. For entities of type *Person*, we set as aliases first names, middle names, and last names. For other types, we find possible aliases using a knowledge base. In DBpedia, there are two properties that are very useful for finding aliases.

1. `dbpedia-owl:wikiPageDisambiguates`: represents the disambiguation page of Wikipedia. For example, by using this property, we know that *"UK"* is an alias for *"United Kingdom"*.
2. `dbpedia-owl:wikiPageRedirects`: stores less common aliases than the first property does, including some frequent typos of the entity, which can be useful in detecting typos to some extent. For example, the Wikipedia *URL* corresponding to *"United Kingdon"* is directed to the Wikipedia page of *"United Kingdom"* and thus we can resolve the misspelling.

[11] http://www.w3.org/RDF/

Once we have all entities with their aliases, we proceed as follows. For each unresolved alias s in comment c, we check if c contains an entity e that has s as alias. If e exists then we replace s by e. Otherwise, we look for e in the parent comment of c. We recursively run this procedure until we find the entity that has s as alias. Formally the algorithm is given as follows.

Data: s: unresolved alias, c: comment containing s
Result: e: entity for which s is an alias
FindEntity(s,c)
begin
  if $c = \emptyset$ **then**
    return null;
  end
  $E \leftarrow getEntities(c)$;
  $e \leftarrow getEntityForAlias(E, s)$;
  if $e \neq \emptyset$ **then**
    return e;
  end
  $FindEntity(s, Parent(c))$;
end

In some cases (although rare), we may encounter an alias that refers to multiple normalized names. In this work, we use a coin toss to decide which normalized name to choose. Considering that *"Rasmussen"* is not extracted as an entity in the previous example, we run our algorithm and if *"'Anders Fogh Rasmussen"* is mentioned in one of the ancestors of the comment, then we replace the string *"Rasmussen"* by *"Anders Fogh Rasmussen"*.

4.3 Coreference Resolution

NER approaches are not designed to detect entities that appear as coreferences. This is a problem for our work since we need to extract aspects related to each entity and thus all types of occurrences of an entity should be taken into account. Let us consider the following example:

> *"If the Scots pull out it leaves the rest of the UK to themselves where the Tories will be the dominant party. They have promised an IN/Out referendum on EU membership."*

From this comment, NER does not recognize the pronoun *"they"* as an occurrence of the entity *"Tories"*. Consequently, *"IN/Out referendum"* and *"EU membership"* cannot be extracted as aspects related to *"Toris"*. To solve this problem we apply the Stanford Deterministic Coreference Resolution System [16] to map coreferences to their corresponding entities. The difference that our work makes is that comments are not independent from each other. In this case, a coreference can be related to an entity mentioned in another comment or in the news article itself. To handle this problem we propose an *Intertextual Coreference Resolution* approach that finds the context of comments to resolve coreferences.

The approach works as follows. For each unresolved coreference, we take the phrase that contains it and append it to the parent of the comment. Then, we apply the Stanford Coreference Resolution[16]. If the coreference is not resolved, we recursively extend the context to the ancestors

until we find the referent or reach the root of the tree. The algorithm of intertextual coreference resolution is formally given by:

Data: r: unresolved coreference, p: phrase containing r, c: comment that gives the context for r, initially set to the parent of the comment where r occurs.
ICR(r,p,c)
begin
  if $c! = null$ **then**
    $context = c.append(p)$;
    $ApplyStanfordCorNLP(context)$;
    if r *is not resolved* **then**
      $ICR(r, p, Parent(c))$;
    end
  end
end

Note that in practice, the referent often belongs to the parent comment or the news article. The reason is that a comment either replies to its parent or comments directly on the content of the news article.

4.4 Context-related Entity Search

Comments might contain aliases that refer to entities which are not mentioned in the news article. If we take the previous example:

> *"Don't be afraid of Rasmussen or NATO, this is none of their business."*

and we suppose that the name "Anders Fogh Rasmussen" is not mentioned in the news article, then applying the name normalization would not work. We call entities mentioned in the news article *explicit entities*, while we call *implicit entities* those that are not mentioned in the news article but brought up by users in their comments. To be able to detect implicit entities, we proceed with a context-related entity search. We start by extracting all explicit entities using a traditional NER tool and the name normalization technique. Further, for each explicit entity, we search the set of entities that are related to it by some arbitrary property in Dbpedia. Note that we use the entity filtering method to remove all implicit entities that do not have the property *rdf:type*. In this work, we do the search only by one step to reduce noise and the running time. However, it might be the case that in the knowledge base implicit entities are connected to explicit entities multiple steps away. The result of this search is then used to extend the entity alias mapping by including the aliases of implicit entities.

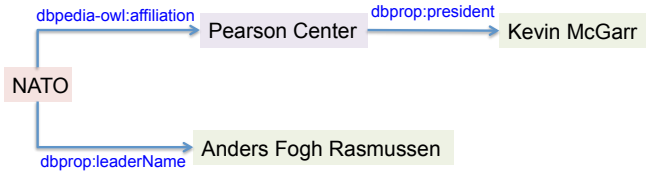


Figure 3: Example of entities extracted by context search

If we take the previous example, we find that the entity *"NATO"* is related to the entity *"Anders Fogh Rasmussen"* via the relationship *dbprop:leaderName* as shown in Figure 3. Note that *"Anders Fogh Rasmussen"* was the Secretary General of *"NATO"*. After finding the entity *"Anders Fogh Rasmussen"*, we set all its possible aliases including *"Anders"*, *"Fogh"* and *"Rasmussen"*. Thus, when *"Rasmussen"* is encountered in a given comment, it is normalized using the name of the implicit entity. In case a comment mentions *"McGarr"*, we need to do a two-step search to find the entity *"Pearson Center"*, then from that entity we find the entity *"Kevin McGarr"* as shown in Figure 3.

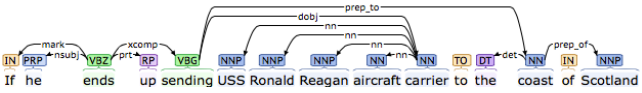
5. ASPECT EXTRACTION

We divide aspects into three categories: (1) *Explicit Aspects*, (2) *Implicit Aspects*, and (3) *Semi-Implicit Aspects*. Explicit aspects appear explicitly as noun phrases, implicit aspects are derived from the context, while semi-implicit aspects are inferred from other part-of-speech tags. In the following, we describe in more detail each type of aspect and present our approach to extracting it.

5.1 Explicit Aspect Extraction

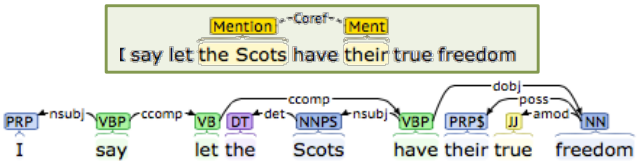
An explicit aspect of an entity e appears as a noun phrase that has a relationship with e. We use the dependency grammar to identify this relationship. Explicit aspect extraction parses comments using the part-of-speech tagging and the dependency grammar, and then selects noun phrases that have a specific grammatical dependency to e. We exploit the following types of dependencies:

Prepositional Dependency. We extract as explicit aspects, for an entity e, all noun phrases that have a relationship with e using prepositions, such as *"of"*, *"at"*, or *"in"*.



In the example above, the aspect *"coast"* is extracted for the entity *"Scotland"* using the propositional dependency *"of"*.

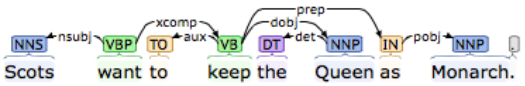
Possessive Dependency. We extract as explicit aspects, for an entity e, all noun phrases that have a possessive dependency to e.



In the example above, the aspect of *"freedom"* is extracted for the entity *"Scots"* by applying possessive dependency, combined with the coreference resolution described in section 4.3. We can see that *"their"* is detected as a coreference for the entity *"Scots"*. Moreover, the noun phrase *"freedom"* has a possessive dependency to the coreference *'their"*. Thus, *"freedom"* has a possessive dependency to *"Scots"*.

Verb Dependency. We extract as explicit aspects, for an entity e, all noun phrases connected to e using a verb that has dependencies to both of the aspects and the entity e. In the following example the aspect *"Queen"* is extracted for the entity *"Scots"* using verb dependency. The clausal

complement of the verb *"want to keep"* has a dependency on both *"Scots"* and *"Queen"*.



The dependencies described above are very precise and thus they rarely fail to extract explicit aspects. However, they only cover a small portion of possible occurrences of aspects in a comment because most of them appear implicitly. Therefore, implicit aspect extraction is needed.

5.2 Implicit Aspect Extraction

To deal with the problem of extracting aspects that are implicit in the text, we propose two approaches. The first approach exploits mappings between adjectives and aspects that co-occur frequently. By contrast, the second approach deals with word-sense disambiguation and lemmatization over implicit aspects.

5.2.1 Adjective-to-aspect Mapping

The extraction rules described in Section 5.1 do not rely on the occurrence of any opinionated word, which makes them different from previous work of explicit aspect extraction [22]. However, the occurrence of an opinionated word is in fact still useful to help extracting some implicit aspects. In this work, we identify the occurrence of words containing sentiment, typically adjectives[12], to build a mapping from pairs of entity and adjective to some explicit aspects.

From all comments, we extract the set of pairs $P = \{\langle e_i, adj_k \rangle\}$ where e_i is an entity, and adj_k is an adjective that has a dependency to entity e_i. To each pair $\langle e_i, adj_k \rangle$, we associate the set of aspects $S_{ik} = \{a_1, ..., a_n\}$ that have a dependency to entity e_i and adjective adj_k. Whenever a pair of entity e_i and adjective adj_k is found in a comment without any occurrence of an explicit aspect, we use adjective-to-aspect mapping. We extract as implicit aspect of e_i, the aspect a_j of S_{ik} which has the highest frequency of occurrence with the pair $\langle e_i, adj_k \rangle$. In the case of multiple aspects that share the value of highest frequency, we use word-sense disambiguation to choose the most appropriate aspect as described in the next section.

> *"Tesco is large! Of course many of the jobs it offers are in shops, transport etc. Those would stay regardless of 'yes' or 'no'."*

In the first sentence of the above comment, *"Tesco"* is an entity and *"large"* is an adjective associated to it. Moreover there is no explicit aspect related to *"Tesco"*. In this case, we check the adjective-to-aspect mapping and we find three occurrences of explicit aspects with the adjective *"large"* and the entity *"Tesco"*: {employer (2), back office (1), call centre operation (1)}. Among these three aspects, we assign to the entity *"Tesco"* the aspect *"employer"*, which has the highest number of occurrences. The aspect *"employer"* was implicitly referred to by the sentence.

[12]We consider adjective to be the only form of the opinionated words, as we do not focus on investigating a sophisticated sentiment analysis.

5.2.2 Lexical Adjective-to-aspect Mapping

Adjective-to-aspect mapping might lead to ambiguity when there are multiple aspects that have the highest frequency with respect to a given entity and adjective. To address this issue, we exploit a lexical database, more specifically WordNet[13] for word-sense disambiguation. If a pair of an entity and an adjective is related to multiple aspects, then we use the similarity measures of WordNet to choose the aspect which is most similar to the context of the entity and its adjective. The context is defined by the sentences around the entity and the adjective. Formally, we define a context c as a set of words $\{w_1, w_2, ..., w_m\}$. The similarity between an aspect a and a context c is computed as follows:

$$Sim(a, c) = \frac{\sum_{i=1}^{m} Wordnet::Similarity(a, w_i)}{|c|}$$

where $Wordnet::Similarity$ is a quantitative measure of the degree to which two word senses are related, provided by WordNet. Basically, for each aspect a we compute the average similarity between a and the words of context c. Then, we choose the aspect with the highest similarity to c.

5.3 Aspect Enrichment

We have seen so far how to extract explicit and implicit aspects from comments. Explicit aspects are extracted from prepositions, possessions, and verbs, while implicit aspects are inferred based on adjectives. An explicit aspect extracted using verb dependency is typically the subject or the object of the verb. By contrast, an implicit aspect is always something that is described by an adjective. Let us take the previous example on verb dependency:

> *"Scots want to keep the Queen as Monarch."*

We note that *"Queen"* is an explicit aspect of *"Scots"* and it is also the object of the verb *"keep"*. Similarly, we take the example of implicit aspects:

> *"Tesco is large!"*

Recall that the implicit aspect *"employer"* was extracted for the entity *"Tesco"*, and in fact this aspect is described by the adjective *"large"*.

The two types of aspects described above can be enhanced with complementary information. The goal is to give more meaning to the aspects discussed about entities. The first comment talks about the aspect *"Queen"* of the entity *"Scots"* and more specifically about *"Keeping the Queen"*. In the same way, the second comment talks about *"Tesco"* being *"large"* and more specifically about being a *"large employer"*. We present in the following our approach for enhancing explicit and implicit aspects.

5.3.1 Explicit Aspect Enrichment

We focus here on aspects extracted using verb dependency. To have complimentary information about an explicit aspect, we convert to a noun the verb to which the explicit aspect is related. In the previous example we convert the verb *"keep"* to *"keeping"*. Then, we concatenate the resulting noun to the explicit aspect, so it becomes *"Keeping the Queen"*.

[13]https://wordnet.princeton.edu

5.3.2 Implicit Aspect Enrichment

To have complementary information about an implicit aspect, we convert to a noun the adjective which is related to the aspect. In the example above, *"large"* is converted to *"largeness"*. Then we concatenate the resulting noun to the implicit aspect, so it becomes *"employer largeness"*.

5.3.3 Semi-implicit Aspect Extraction

In some cases, entities are not related to any explicit or implicit aspect, but they are related to an action or described as a whole without referring to a specific side. Let us take the following example:

> *"Scotland can vote however it wants, it's the Scottish peoples right."*

The first sentence of this comment talks solely about *"Scotland"* voting. Consider another example:

> *"I would love to visit Scotland again—hopefully an independent one—it is a very beautiful country and the Scots are so honest."*

Here we can see that the entity *"Scots"* as a whole is described as honest. To generalize, if an entity (1) has a dependency to a verb and (2) the verb is not related to an explicit or implicit aspect, then we convert the verb to a noun and use it as an aspect for the entity. The same thing applies for adjectives. Applying this rule, we would have from the two previous examples the aspect *"voting (or vote)"* assigned to *"Scotland"* and the aspect *"honesty"* assigned to *"Scots"* .

To convert verbs and adjectives into nouns, we use WordNet, which returns for each word a list of nouns that are lexically related. There are different relations that can be exploited such as *"synonym"*, *"similar to"*, and *"derivationally related form"* to find the noun form of a verb or an adjective. Among the returned nouns, we choose the one most similar to the input, using the *Wordnet::Similarity* function.

6. EXPERIMENTS

In this section, we describe the setup of our experiments, then we present and discuss the results.

6.1 Setup

Dataset. We have crawled comments on 10 different news articles from the Al Jazeera and CNN websites. For each news article, we have selected the 100 longest comments. For each comment we have added the set of its ancestor comments to keep track of its context. This process resulted in 1087 comments on the 10 news articles. We have chosen the articles from three different categories: *Politics*, *Sport*, and *Techs*. The reason of this choice is to reflect the various types of user comments. Comments on sport category have a high tendency to introduce entities that are not mentioned in the article. Comments on techs category have the same characteristic of sport comments and in addition they have the tendency to use slang language. Finally, comments on politics are highly controversial and discuss a variety of aspects regarding the same entity. Besides the ancestor comments, the distribution of comments on the three categories is: 400 comments in politics, 300 comments in techs, and 300 in Sport

Annotation. The collected dataset of 1087 comments was annotated by first-year bachelor students who are not involved in this project. Their task was to extract from each comment the set of entities and aspects using the following guidelines:

1. An entity is either a person, an organisation, a location, or a well-defined concept (non arguable or subjective) such as languages.
2. An explicit aspect is a noun phrase that has a grammatical dependency on an entity.
3. An implicit aspect is referred to by an adjective that has a dependency to an entity.
4. A semi-implicit aspect is the noun form of a verb or adjective that have a dependency to an entity

It is important to mention that the students had the freedom to extract the implicit aspects based on what they understood from the comment, decide the suitable noun form of the verb or the adjective, and decide whether something is a well-defined concept or not. The results of these annotations were use as our ground truth data.

Baselines. We have used three baseline approaches for entity extraction: NERD and Zemanta, which are two of the best NER tools [17, 18], and AIDA, which has been demonstrated to have an accuracy competitive to the very best named-entity disambiguation (NED) systems [4]. We have used the baseline approaches to extract entities from user comments. Then, we applied the different approaches we have proposed to assess their impact on the results. Regarding aspect extraction, there is no existing work that is really comparable to ours, because all work on aspect extraction has mainly focused on product reviews[15]. More importantly, they assume the input text to be always about a single entity, whereas texts in our dataset may contain multiple entities. Thus, they perform poorly.

Metrics. To assess the performance of our approach, we use two evaluation metrics: *precision* and *recall*. Precision is given by:

$$Precision = \frac{truePositives}{truePositives + falsePositives}$$

and recall is given by:

$$Precision = \frac{truePostives}{truePositives + falseNegatives}$$

6.2 Results

We have tested separately entity extraction and aspect extraction techniques and the results are described in the following.

6.2.1 Entity Extraction

The overall results are shown in Table 1. We can see the approach we have proposed for extending existing NER improves substantially both precision and recall. We note that precision is improved by 22% for Zemanta, and 30% for NERD. Similarly, the recall is improved by 34% for Zemanta and 25% for NERD. Although we slightly improved the precision of AIDA, we have significantly improved its recall by 15% mainly by detecting more forms of occurrences of entities. More details about the impact of each proposed technique to enhance existing NER are shown in Tables 2, 3,

	Underlying NER		Our Proposed Extensions	
	Prec.	*Recall*	*Prec.*	*Recall*
NERD	56.94	52.13	**86.18**	**77.50**
Zemanta	72.74	50.35	**90.09**	**74.94**
AIDA	81.67	56.93	**82.08**	**72.34**

Table 1: Overall Precision and Recall (in percent) for Entity Extraction

and 4. They report the overall precision and recall together with the average values for each category of user comments including politics, techs, and sport. The different techniques we have proposed, in this paper, are applied incrementally from the leftmost one (traditional NER tool as baseline) to the rightmost one (coreference resolution). Therefore, the numbers listed in the coreference resolution are obtained after applying all the techniques in order. The results for each technique are presented in the following.

Entity Filtering. After applying entity filtering, the precision of all baselines is boosted for a little drawback on recall. If we use Zemanta, the entity filtering increases the precision by 27%, but reduces the recall by 0.5% for all categories, while when we use NERD, the precision increases by 64% and the recall is reduced by 0.4%. Note that the precision of NERD increases more than the one of Zemanta because NERD produces more noise. Regarding AIDA, it returns as entities instances of classes thus the improvement is very small.

Name Normalization. The improvement that name normalization brings is on the recall. This is natural since its goal is to detect more occurrences of entities. While precision values are almost the same for entity filtering and after applying name normalization, recall has an overall increase of 4% for both Zemanta and NERD and less than 3% for AIDA. We note that the increase depends on the category of comments. For techs, it is around 4% and for politics is two times higher. This is due to the fact that typically techs entities are not mentioned using aliases while in politics they are. We note that AIDA behaves differently because it already performs labeling people, organization, and country names in abbreviation. However, we provide an additional improvement of 3% on its precision and recall values

Context Search. The same observation holds for context search which improves the recall keeping the values of precision stable. We can note an additional improvement of 11% in overall recall for both Zemanta and NERD. It is interesting to see that the improvement depends on the category of comments. For example, in the techs category, the recall improves by 2% at most while in sports is of 30%. This huge difference is due to the fact that sport comments often talk about entities that are not mentioned in the news article. For example, a comment may mention members of a football team while the article talks on the team as a whole, without using specific names. In this case, context search is more evident helps extracting these implicit entities and thus we can see the impact clearly in sport comments. This strategy penalizes slightly the precision which is natural, since we collect from the knowledge base extra information that can contain some noise. However, the decrease in precision is negligible. Although AIDA performs also context search, our approach provides additional improvement of 4% on the recall.

Coreference Resolution. After applying coreference resolution, we complete the integration of all the methods we have proposed in the entity extraction process. We provide an additional improvement in recall of more than 9% for Zemanta, NERD, and AIDA while we have a slight decrease in precision of less than 3%. This is because coreference resolution extracts more occurrences of entities, but sometimes it infers the wrong entity which penalizes the precision.

6.2.2 Aspect Extraction from Clean Data

In this section, we present the experiment results for our aspect extraction approach based on our annotated dataset. This means that we take the annotated entities as starting point. The reason is to analyze the performance of our approach independently from the influence of external tools. The results are shown in Table 5. We can see that the combination of explicit, implicit, and semi-implicit techniques provide the best trade-off between precision and recall. The details of the performance of each technique are presented in the following.

Explicit Aspect Extraction. This technique achieves a very high precision of 90.88% because the extraction rules are designed to correctly capture all explicit aspects. However it fails, in very few cases, when figurative language is used. For example, consider the sentence *"This game is a piece of cake for Arsenal?* Our approach will extract *"piece of cake"* as an aspect of *"Arsenal"* because of the presence of a prepositional dependency between them, while in fact, the term *"piece of cake"* is an idiom that suggests an aspect of *"easiness"*. In contrast, the recall of this technique is very low because most aspects appear implicitly.

Explicit + Implicit Aspect Extraction. We can see in Table 5 that adjective-to-aspect mapping is not very useful when tested on our annotated data. It increases the recall by less than 3% while it decreases the precision by 14%. The reason is that choosing implicit aspect based on the highest frequency of occurrences may create a bias which negatively influences precision and recall. However, when we use the extended adjective-to-aspect mapping, we improve both precision and recall values because the use of aspect disambiguation decreases the likelihood of bias.

Explicit + Semi-implicit Aspect Extraction. Using semi-implicit aspect extraction together with explicit aspect extraction increases the recall from 23.50% to 65.62%, which is substantial. The precision is deceased because of the inaccuracies of choosing the best aspects using the lexical relations and similarity measures. The *Wordnet::Similarity*

	Zemanta (baseline)		+Entity Filtering		+Name Normalization		+Context Search		+Coreference Resolution	
	Prec.	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*
Politics	70.01	59.17	89.33	58.83	89.23	67.43	89.22	71.99	89.07	81.31
Techs	74.10	52.33	94.52	52.31	94.51	53.09	94.51	54.98	89.43	61.62
Sport	75.04	36.61	96.11	36.61	96.09	40.68	95.89	70.61	92.08	79.81
Average.	72.74	50.35	92.92	50.21	92.88	55.06	92.81	66.75	90.09	74.94

Table 2: Precision and Recall (in percent) for Entity Extraction based on Zemanta

	NERD (baseline)		+Entity Filtering		+Name Normalization		+Context Search		+Coreference Resolution	
	Prec.	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*
Politics	52.21	61.21	86.47	60.95	86.39	67.74	86.38	73.66	86.19	83.58
Techs	59.43	54.18	90.48	54.16	90.47	55.01	90.46	55.97	85.70	63.26
Sport	60.27	37.96	90.52	37.95	90.50	42.16	90.15	73.18	86.69	83.91
Average	56.94	52.13	88.90	52.01	88.86	56.17	88.75	68.13	86.18	77.50

Table 3: Precision and Recall (in percent) for Entity Extraction based on NERD

itself is not optimized for comparing similarity between synsets other than nouns and verbs [13], therefore the drawback in precision is understandable.

6.2.3 Aspect Extraction from Noisy Data

To assess the effectiveness of our aspect extraction approach, we use the set of entities provided by the baseline approaches after applying the extension techniques as shown in section 6.2.1. The set of entities would contain noisy information and the goal is to analyze the impact of this noisy information on the precision and recall of aspect extraction. The results are shown in Table 6. We can observe that using NER tools decreases both precision and recall compared to ground truth data which is obvious. The interesting observation is that AIDA is the one performing worse causing a decrease of 12% in precision and 22% in recall. The best in terms of precision is Zemanta, and the best in term of recall is NERD. These results are very consistent with the entity extraction findings where AIDA is the worst, while Zemanta and NERD provide the best precision and recall values.

	Prec.	Recall
Ground Truth	73.42	73.82
Zemanta	66.33	56.37
NERD	63.36	58.48
AIDA	60.69	55.63

Table 6: Aspect Extraction from noisy data

6.3 Discussion

We have shown that our proposed techniques for extending NER tools improve significantly precision and recall values. Interestingly, AIDA performs best when we do not apply any extension. This is because it already provides similar strategies to what we have proposed for entity filtering, name normalization, and context search. However, Zemanta and NERD, which are relatively much simpler, perform better than AIDA when we extend them with our proposed techniques. There are two benefits from this result: (1) we do not depend on one NER tool but can use different tools to cover diverse types of entities while still guaranteeing high quality results; and (2) we can improve the performance of a simple NER tool using our light-weight extension techniques and outperform more sophisticated tools such as AIDA. The simplicity of our approach makes the results more effective and the computation time faster.

Another important point to consider is that AIDA uses Stanford NER which is supervised, while NERD and Zemanta are unsupervised. In our work, we aim at giving more room to unsupervised techniques because in news domain, new types of entities continue to emerge and we want to have a system that can handle that. For example, AIDA does not detect entities of type product, so we need to be able to use another NER to extract this type of entities.

Regarding aspect extraction, we have seen that we achieve 73% on recall and precision on the ground truth data. This result is very encouraging. We can also see that if the input is noisy, aspect extraction provides less accurate results because of the decrease in recall and precision of extracted entities. This is also due to some specific reasoning that NER tools perform on entities, such as AIDA that labels *"Scotland"* as *"United Kingdom"*, *"ISIS"* as random entities, and *"Ukraine* as Russian Empire.Inaccurate labelling would then lead to inaccurate aspects.

7. CONCLUSIONS

We have presented, in this paper, a new approach for entity and aspect extraction from user comments. We have introduced light-wight techniques to extend existing NER tools using coreference resolution and comment context information. We have defined new forms of aspects that can be either explicit or implicit and proposed extraction techniques for each type. The experiments results have shown that our approach is promising giving new insights about information extraction from user-generated content. In future work, we aim at testing our approaches on larger datasets, which is a tedious task because of manual annotation. Furthermore, we need to investigate aspect extraction techniques to improve their performance.

	AIDA (baseline)		+Entity Filtering		+Name Normalization		+Context Search		+Coreference Resolution	
	Prec.	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*	*Prec.*	*Recall*
Politics	77.82	66.35	78.08	66.34	80.21	67.88	80.17	68.51	80.01	77.56
Techs	83.51	69.33	83.75	69.33	88.37	75.75	87.33	76.07	83.30	85.98
Sport	91.55	31.97	91.89	31.95	91.89	32.44	90.86	45.64	86.67	51.74
Average	81.67	56.93	81.93	56.92	84.73	59.61	84.19	63.92	82.08	72.34

Table 4: Precision and Recall (in percent) for Entity Extraction based on AIDA

	Precision	Recall
Explicit	**90.88**	23.50
Explicit + Implicit (frequent mapping)	76.87	26.12
Explicit + Implicit (lexical mapping)	88.65	30.03
Explicit + Semi-Implicit	72.42	65.62
Explicit + Implicit+ Semi-Implicit	73.42	**73.82**

Table 5: Overall Precision and Recall (in percent) for Aspect Extraction

8. REFERENCES

[1] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Conference of the North American. ACL, 2003.*

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[3] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *5th conference on Applied natural language processing*. ACL, 1997.

[4] C. Bizer, T. Heath, S. Auer, and T. Berners-Lee, editors. *Workshop on Linked Data on the Web (WWW)*, 2014.

[5] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *The Sixth Workshop on Very Large Corpora*, 1998.

[6] H. L. Chieu and H. T. Ng. Named entity recognition: a maximum entropy approach using global information. In *The 19th international conference on Computational linguistics*. ACL, 2002.

[7] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *The 43rd Annual Meeting on Association for Computational Linguistics*. ACL, 2005.

[8] M. Hu and B. Liu. Mining and summarizing customer reviews. In *ACM SIGKDD*, 2004.

[9] D. Kitayama, N. Oda, and K. Sumiya. Organizing user comments in a social video sharing system by temporal duration and pointing region. INGS'08, 2008.

[10] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. *WWW*, 2005.

[11] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *7th Conference on Natural language learning at HLT-NAACL*. ACL, 2003.

[12] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 2007.

[13] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*. ACL, 2004.

[14] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper. Topic discovery based on text mining techniques. *Information processing & management*, 2007.

[15] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh. A rule-based approach to aspect extraction from product reviews. *SocialNLP 2014*, page 28, 2014.

[16] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A multi-pass sieve for coreference resolution. In *The 2010 Conference on Empirical Methods in Natural Language Processing*. ACL, 2010.

[17] G. Rizzo and R. Troncy. Nerd: evaluating named entity recognition tools in the web of data. In *Proceedings of ISWC*, 2011.

[18] G. Rizzo, R. Troncy, S. Hellmann, and M. Bruemmer. Nerd meets nif: Lifting nlp extraction results to the linked data cloud. *LDOW*, 937, 2012.

[19] S. Sekine, R. Grishman, and H. Shinnou. A decision tree method for finding and classifying names in japanese texts. In *6th Workshop on Very Large Corpora*, 1998.

[20] S. Sekine and C. Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, 2004.

[21] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. AIDA: an online tool for accurate disambiguation of named entities in text and tables. *PVLDB*, 4(12):1450–1453, 2011.

[22] L. Zhang and B. Liu. Aspect and entity extraction for opinion mining. In *Data Mining and Knowledge Discovery for Big Data*, Springer, 2014.