Harmony and Dissonance: Organizing the People's Voices on Political Controversies

Rawia Awadallah
Max-Planck Institute for Informatics
Saarbruecken, Germany
rawadall@mpi-inf.mpg.de

Maya Ramanath
Indian Institute of Technology, Delhi
New Delhi, India
ramanath@cse.iitd.ac.in

Gerhard Weikum
Max-Planck Institute for Informatics
Saarbruecken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

To understand the landscape of opinions on political controversies, it would be helpful to know which politician or other stakeholder takes which position - support or opposition - on specific aspects of these topics. We present a system, named OpinioNetIt, which aims to automatically derive a map of the opinions-people network from news and other Web documents. We build this network in four stages. First, we start with a few generic seeds to identify phrases denoting controversial topics from text. These phrases are then automatically organized and mapped into a DAG of topics. Second, opinion holders are identified for each topic, and their opinions (support or oppose) are extracted. Third, the acquired topics and opinion holders are used to construct a lexicon of phrases indicating support or opposition. Finally, we iterate this process, using the richer lexicon to identify more opinion holders, opinions and topics. We present a systematic evaluation which shows the high accuracy of OpinioNetIt. We also present use-case scenarios for identifying flip-flop politicians who change their opinions on the same topic, and for discovering people with divergent opinions on otherwise highly correlated topics.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic Processing*

General Terms

Algorithms, Experimentation

Keywords

Web Information Extraction, Political Opinion Mining

1. INTRODUCTION

1.1 Motivation

The crises in Libya and Syria, the debates about the economic crisis in Greece, and the downrating of the USA's creditworthiness are some of the controversial topics being played on the news everyday. Each of these topics has many different aspects, and there is no absolute, simple truth in answering questions such as: should the EU guarantee the financial stability of each member country, or should the countries themselves be solely responsible? To understand the landscape of opinions, it would be helpful to know which politician or other stakeholder takes which position - support or opposition - on these aspects of controversial topics. In order to navigate and analyze this complicated landscape of issues and opinions, it would be helpful if there were an *"opinion-base"* that organized controversial topics according to their various facets and stored information about who expressed an opinion on one or more topics and supported or opposed a certain direction.

This envisioned network goes beyond state-of-the-art sentiment mining in several ways. First, political controversies are much more complex and opinions are often expressed in subtle forms, which makes determining pro/con polarities much more difficult than with product reviews for cameras, movies, etc. - the typical objects in prior work on opinion mining [16, 13]. Second, most prior research focused on classifying individual sentences or reviews into pro/con categories or on aggregating (summarizing) opinions over a large number of observations (e.g., many different customers' reviews of some product). In contrast, our goal is to connect an opinion to an individual person (typically a politician) who expressed this position (multiple times, but often in very different wordings). Third, instead of merely finding a few or the most interesting pairs of opinion holders and topics, we aim at a systematic network of opinion holders and pro/con statements for a wide variety of fine-grained controversial topics. To the best of our knowledge, no prior work has addressed political opinion analysis in a similarly comprehensive manner.

Facet	Canonical Name
use of force against civilians in Libya	Muammer Gaddafi's response to the Libyan civil war
using military force against the regime of the Libyan leader	Military Action in Libya
No-Fly Zone over Libya	Military Action in Libya

Table 1: Example facets extracted for "Libyan Civil War"

⟨Nicolas Sarkozy⟩ ⟨support⟩ ⟨Military Action in Libya⟩
⟨Dmitry Medvedev⟩ ⟨oppose⟩ ⟨Muammer Gaddafi's Respponse to the Libyan Civil War⟩
⟨Angela Merkel⟩ ⟨support⟩ ⟨Military Action in Libya⟩

Table 2: Opinions about "Libyan Civil War"

In this paper, we present our system, called *OpinioNetIt* (pronounced similar to "opinionated"), which aims at building such an opinion-base for controversial topics by extracting and organizing information from online sources on the Web. For example, the first

WSDM'12, February 8–12, 2012, Seattle, Washingtion, USA.

column of Table 1 shows some of the facets extracted by our system regarding the "Libyian Civil War". The outcome of analyzing these inputs should be a crisp set of structured records of the form: *person* (opinion holder), *polarity* (support/oppose), and *topic facet* (fine-grained controversial topic). Table 2 shows examples of people's opinions about various topic facets of the controversy. For example, ⟨Angela Merkel⟩ ⟨support⟩ ⟨Military Action in Libya⟩ could be a result in canonicalized form. This desired output involves aggregating statements, but only for the same pair of opinion holder and topic facet. The overall set of such records forms the OpinioNetIt network.

The challenge in building this network lies in the fact that the input is merely natural-language text, such as news articles, where it is difficult to spot phrases that denote individual politicians or controversial topics and map them into a canonical representation. For example, the opinions of Table 2, are extracted from text where the entity Vladimir Putin is mentioned several times with different wordings like "Mr. Putin", "the Russian President", "President Putin"etc. The same language diversity and ambiguity applies to the topic facets that we aim to detect and organize. Thus, the entire problem can be seen as an interleaved task of fact extraction, entity resolution, and opinion assessment.

1.2 Approach and Contribution

OpinioNetIt currently taps into 5 online news sources: Google News, Al-Jazeera, BBC, CNN, New York Times, and Wikipedia, and extracts 4-tuples of the following form:

⟨person⟩ ⟨support/oppose⟩ ⟨topic⟩*{⟨context⟩}*

where the triple: ⟨person⟩ ⟨support/oppose⟩ ⟨topic⟩ is extracted from the textual *context*.

We specifically choose the triple+context format since it is compatible with the RDF data model and recent efforts to extend RDF with contextual information [6]. The advantage of using RDF is that SPARQL, the query language for RDF, can be used to query the system. Queries of the form: "who opposes the military action in Libya", "who supports Muammer Gaddafi's respponse to the Libyan rebels", etc. can now be processed by the system.

Although our input is currently limited to 5 sources, our methods easily extend to additional Web sources. Our approach is to generate search-engine queries with site scope restricted to each of our sources, and extract opinions from the snippets returned by the search engine. The extraction is carried out in the following stages (see Figure 1 for an overview).

1. Acquisition and Organization of Topics: First, we manually construct a small list of seed patterns consisting of phrases denoting support and opposition. We query the Web with these seeds and acquire result snippets from the 5 sources listed previously. We then parse the snippets and extract phrases which correspond to facets of a controversial topic. Once we acquire a large number these phrases, we map them onto canonical topics and organize these in a topic hierarchy.

2. Acquiring Opinion Holders: Second, from the collected snippets, we identify opinion holders by deep natural-language parsing and judiciously designed heuristics. One of the assets harnessed here is the YAGO knowledge base [18] which provides us with many name variants and paraphrases of individual entities.

3. Building the Support/Oppose Lexicon: Third, given the facet and an opinion holder, we again query the Web and build a lexicon of support and oppose phrases. Some examples of such phrases are: "strongly against" (for oppose) and "acknowledged the validity of" (for support), etc.

4. Gathering More Opinions: Finally, by using the lexicon of support/oppose phrases as new seeds, we close the loop: we go back to the first step and iteratively acquire more topics and opinion holders, thus building the final OpinioNetIt network.

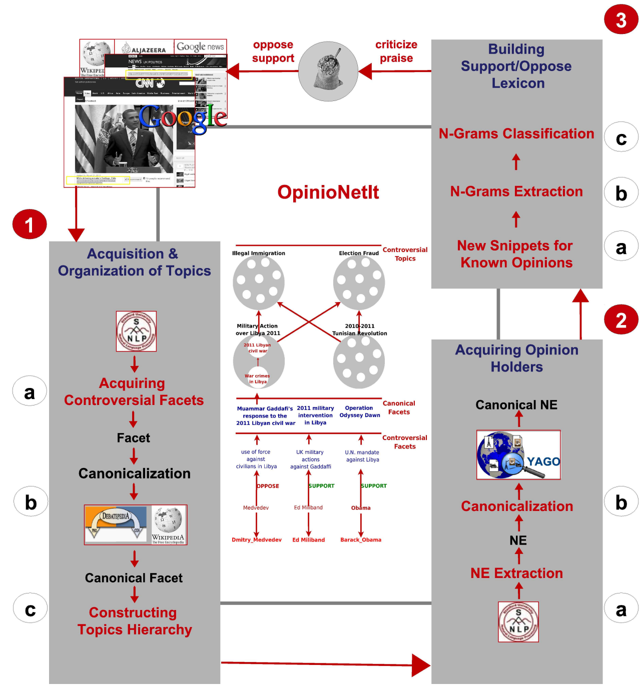

Figure 1: System architecture.

The contribution of this paper is the methodology for constructing this new style of opinion-base, like OpinioNetIt, which consists of controversial topics and their various facets and the people who hold opinions on it. We restrict our opinions to be of two types: support and oppose. Our methodology for achieving this makes use of a variety of building blocks and combines them in an innovative way. In particular, we integrate techniques for pattern-based information extraction with polarity classification, both customized to our setting of fine-grained political controversies. We present an experimental evaluation of our methods, and show that we achieve an overall precision of 72% on the extracted opinions. We also present experimental comparisons for alternative algorithms used in critical components (e.g. organizing facets and topics). Finally, we show use-case scenarios for identifying flip-flop politicians who change their opinions on the same topic, and for discovering people with divergent opinions on otherwise highly correlated topics.

The rest of the paper is organized as follows. Section 2 introduces our computational model. The different stages of our approach: acquiring and organizing controversial facets is in Section 3, identifying opinion holders is in Section 4, and building the support/oppose lexicon is in Section 5. The system implementation is described in Section 6. Section 7 presents our experimental evaluation, and Section 8 discusses use cases. Section 9 positions our contributions with regard to related work, and we conclude with Section 10.

2. COMPUTATIONAL MODEL

Our first major goal is to acquire politically controversial topics and to organize them into a hierarchy. For example, both "sanctions against Iran" as well as "military strikes on Iran" are different *facets* of the debate on "Iran's nuclear program" which in turn can be regarded as being a part of a larger debate on "nuclear power". Similarly, "Japan's nuclear disaster" is also part of the debate on "nuclear power". Constructing such a hierarchy, however, presents challenges. First, we have to acquire facets which are politically controversial. Second, since each facet can be expressed in different ways, we have to *canonicalize* these facets. That is, "military strikes against Iran" and "military action on Iran" are different ways of expressing the same topic and need to have a canonical form. Third, once we have canonical forms we need to identify the topics to which they belong. Before we describe each of these components in detail in Section 3, we first explain more precisely what we mean by terms such as "topic", "facet", etc. and formally define our goal.

A *topic* is informally defined as the subject matter of a particular piece of text. A *controversial topic* is a topic on which drastically varying opinions exist among people. A topic can be divided into a number of *subtopics* each of which can themselves be divided further. At the most basic level, we deal with *facets* (also referred to as *raw facets*) of a topic (or subtopic) and consider facets to be indivisible. Intuitively, the set of topics, subtopics and facets form a hierarchy.

With this understanding of the terms *topic*, *subtopic* and *facet*, we now introduce some formal terminology.

DEFINITION 2.1. *A* **facet** f_i *is denoted by a string* $str(f_i)$*. Examples of facets on the controversial topic "Iran's Nuclear Program, 2010" include "military strikes on Iran" and "sanctions on Iran". We use the term* facet name *to refer to* $str(f_i)$.

DEFINITION 2.2. *A* **canonicalized facet** F_i*, denoted by a string* $str(F_i)$ *corresponds to a set of facets* $S(F_i) = \{f_1, f_2, \ldots, f_k\}$ *where each* f_j *has the same semantic meaning. For example, "military strikes on Iran" and "military action against Iran" are both semantically the same and we may choose the former as the canonical form. Unless mentioned otherwise, we simply use the term "facet" to mean "canonicalized facet". We also use the term* canonicalized facet name *to refer to* $str(F_i)$.

DEFINITION 2.3. *A* **topic** $T = \{Q_1, Q_2, \ldots, Q_n\}$ *where* Q_i *denotes a facet or a subtopic. As an example, "Nuclear Proliferation" may have subtopics such as "Iran's Nuclear Program, 2010", "Nuclear Non-Proliferation Treaty", etc. And "Iran's Nuclear Program, 2010" in turn has facets such as "military strikes on Iran" and "sanctions on Iran". We denote the set of* all *topics, subtopics and facets by* $\mathcal{T}$.

DEFINITION 2.4. *An* **opinion holder** P *for a topic* T *or a facet* f *is an entity such as a person, organization or government that holds an opinion on* T *or* f*. We denote the set of all opinion holders as* $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$.

DEFINITION 2.5. *A* **topic hierarchy** *is a DAG consisting of topics, subtopics and facets. Formally, let* $TH = \{V, E\}$*, where* $V = \{v_i | v_i \in \mathcal{T}\}$ *and* $E = \{(v_i, v_j) | v_j$ *is a subtopic or facet of* $v_i\}$.

DEFINITION 2.6. *An* **opinion fact** *is denoted by:*
⟨*P*⟩ ⟨*support/oppose*⟩ ⟨*Q*⟩{⟨context⟩}
where the triple: ⟨*P*⟩ ⟨*support/oppose*⟩ ⟨*Q*⟩ *is extracted from the textual* context *and* $P \in \mathcal{P}$ *denotes the entity holding an opinion on* $Q \in \mathcal{T}$.

DEFINITION 2.7. *An* **opinion network** *is a node- and edge-labeled, directed graph* $ON = (V, E, C)$ *where* $V = \{v_i | v_i \in \mathcal{T} \cup \mathcal{P}\}$ *and* $E = \{(v_i, v_j, l) | v_i \in \mathcal{P} \cup \mathcal{T}, v_j \in \mathcal{T}, l \in \{$*support, oppose,* $\epsilon\}\}$ *and* $C : E \rightarrow \{str_1, str_2, ..., str_n\}$ *maps each edge to a context.*

The topic hierarchy TH *is part of the opinion network (therefore, we have edges with both endpoints in* $\mathcal{T}$ *and with empty labels). Note that the* triples *in* opinion facts *each correspond to a labeled edge in the graph and the mapping* C *associates each triple to its context.*

3. FACETS AND TOPICS

In this section, we describe our solutions to the problem of acquiring politically controversial topics and organizing them into a hierarchy.

3.1 Acquiring Controversial Facets

Our aim is to collect a large number of opinions (high recall). While using Wikipedia was an option we considered, we felt that comprehensive coverage of topics would be possible only on the Web. Therefore, we decided to use the Web to gather a large number of topics and opinions, while Wikipedia was used as a resource to help organize the topics.

Our initial approach to acquiring a candidate set of controversial facet used only two seed patterns: "support" and "oppose". These two patterns were fired as queries to a search engine and the resulting snippets were gathered. We parsed the individual sentences in each snippet using the Stanford parser [12] in order to identify phrases corresponding to facets. However, this initial approach proved insufficient for two reasons. First, simple "support" and "oppose" are not specific enough to i) return only political topics, and ii) return topics on which a *person* (or organization) could have an opinion. Second, simply firing these queries on the Web often returns snippets which are very difficult or even impossible to correctly parse (e.g., user comments in forums).

In order to solve the first problem, we refined our seed patterns. To ensure that the facets returned were those for which people could have opinions, we used the patterns "he supports" and "he opposes" (and their female counterparts). In order to keep the focus on political topics, we added the patterns "he voted for", "he voted against" on the intuition that most "voting" typically takes place in political forums. For the second problem of bad snippets, our pragmatic approach is to still query the Web, but limit the sites from which results are obtained. Our current sources are: Google News, Al-Jazeera, BBC, CNN, New York Times, and Wikipedia.

3.2 Canonicalization of Facets

Once the raw facets are collected, we organize them in two ways. First, in order to ensure uniformity in referring to semantically equivalent facets (e.g., "financial aid for Greece" and "EU loan to Greece" are equivalent), we automatically derive canonical names. Second, for convenient exploration and knowledge discovery at different granularities, we impose a topic hierarchy on the canonicalized facets. This subsection discusses our approach to the first issue, the second issue is addressed in Subsection 3.3. Figure 2 shows an example of what the final hierarchy looks like in the context of the recent Libyan crisis. Some of the raw facets we acquired are shown in blue (bottom level of the hierarchy) and include a variety of facets: "U.N. Mandate against Libya", "use of force against civilians in Libya", etc. These raw facets are then canonicalized to "2011 military intervention in Libya" and "Gaddaffi's response to the 2011 Libyan Civil War" respectively which in turn are part of the larger topic "2011 Libyan Civil War".

For canonicalizing facets, we devised a *two-phase mapping technique*, based on Wikipedia and Debatepedia (`debatepedia.org`). Debatepedia consists of ca. 1700 debates organized into ca. 1300 topics such as Iranian nuclear crisis, Abortion, Homosexuality, etc. Each topic consists of a number of debates – for example, "Abortion" has debates on "Women's rights", "Fetus' rights" "Birth control", etc. The rationale for using both Debatepedia and Wikipedia is as follows. Debatepedia is focused on political controversies but not that large, whereas Wikipedia is much richer but controversial topics are only a small part of it. We use Debatepedia as a bootstrapping asset to focus on controversies, but eventually map facets to Wikipedia categories for a richer organization of topics.

Mapping Phase 1: To ensure that our facets are indeed of a controversial nature, we first build a classifier that maps facets onto Debatepedia debates (rather than trying to map directly to Wikipedia). To this end, we have built a nearest-neighbor classifier that uses statistical language models (LM) (an IR technique) as the basis for its distance measure. We also considered alternative classifiers like Bayesian or SVM models, but the emphasis here is on the feature space and the kNN method works very well.

DEFINITION 3.1. *Let $T = \{T_1, T_2, \ldots, T_m\}$ be the set of topics in Debatepedia and let D_T denote the set of all documents debating the topics in T. Recall that each topic has many different facets that may be debated. Let D_i be the set of documents which debate the various facets of the topic T_i. The* language model for a debate D_i*, smoothed with all debates D_T, is the following probability distribution over words (or phrases):*

$$P_{D_i}(w) = (1 - \lambda)P(w|D_i) + \lambda P(w|D_T)$$

where w is a word, $P_{D_i}(w)$ is the estimated probability of the word in the LM of D_i, $P(w|D_i)$ is the probability of the word in D_i and $P(w|D_T)$ is the probability of w in the "background corpus" D_T, consisting of all debates in T, and λ is a Jelinek-Mercer-style smoothing coefficient (or derived from a Dirichlet smoothing model).

We now map a raw facet f onto its nearest debate D (treating f as a query in LM-based IR terminology [8, 22]): that is, the D with maximum likelihood of generating f.

DEFINITION 3.2. *The* **ranking of debates** D_i *for a given facet f is in descending order of*

$$P(f|D_i) = \prod_{w_j \in f} P_{D_i}(w_j)$$

We compute the mean of the LM scores of the top-5 debates, and the mean of the LM scores of the bottom-5 debates. If the two means are statistically different using the T-test at $\alpha = 0.05$, the top-ranked debate is chosen as a candidate name for a canonical facet. Otherwise facets are discarded. We use the same heuristic for the Wikipedia articles ranking and for the Wikipedia categories ranking which we will describe later in this section.

Mapping Phase 2: Since, Debatepedia is smaller than Wikipedia, we extend the set of debates by mapping each debate to the 3 most similar articles in Wikipedia, again using an LM-based kNN method. The titles of these articles serve as candidates for canonical names of raw facets. For example, the recent Libyan civil war is covered in "No-fly zone over Libya", "Human rights violation in the 2011 Libyan civil war", and "National transitional council". These titles are different facets of the same topic.

For finding the 3 best Wikipedia articles, we would like to avoid LM comparisons with a large number of articles. Therefore, we first derive a Google query from the debate D and restrict the search results to Wikipedia. Since the entire text of D would be unsuitable as input to a search engine, we generate the query from the words in the title and 10 most frequent noun phrases in D. The top-10 search results $A_1, A_2, \ldots, A_{10}$ (corresponding to Wikipedia articles) are our candidates for the D-to-A mapping.

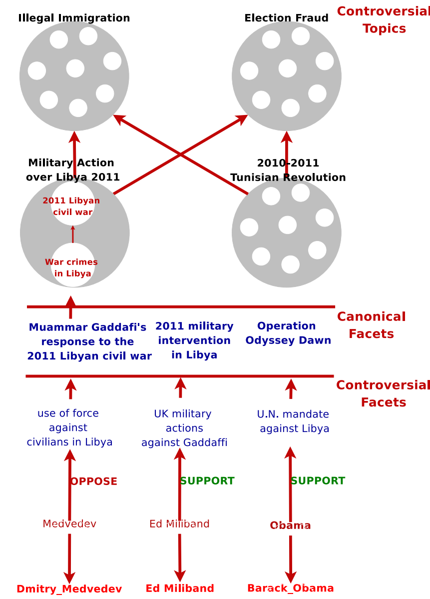

Figure 2: Organizing facets in different levels of granularities.

DEFINITION 3.3. *The* **language model for Wikipedia article** *A, smoothed with the entire Wikipedia as a background corpus W, is the probability distribution over words (or phrases):*

$$P_A(w) = (1 - \lambda)P(w|A) + \lambda P(w|W)$$

The **ranking of articles** A_i *for a given debate D is in descending order of*

$$P(D|A_i) = \prod_{w_j \in D} P_{A_i}(w_j)$$

3.3 Constructing the Topic Hierarchy

Once all raw facets have been mapped to canonical facet names and are associated with 3 Wikipedia articles, the next task is to organize them into a hierarchy of topics. A seemingly natural approach would be to use the Wikipedia category system. However, a major problem with such an approach is that not all categories to which an article belongs are really of controversial nature (e.g., "History of Greece", "European Union", etc.).

To make this important distinction, we devise the following approach. First we collect all Wikipedia categories associated with the 3 articles to which a facet F and its corresponding debate D were mapped, into candidates pool $C_w = \{c_{w_1}, c_{w_2}, \ldots\}$. Second, we use the Debatepedia categories $C_d = \{c_{d_1}, c_{d_2}, \ldots\}$ of the debate D to generate Google queries, restricted to Wikipedia categories (analogous to to the technique in Subsection 3.2). For each c_{d_i}, we obtain top-10 Wikipedia category pages but add only the 3 highest ranked ones to the candidates pool C_w . This step serves to reduce the candidate space. Subsequently, we employ LM-based kNN mapping to obtain the best 3 Wikipedia categories from C_w for the debate D.

DEFINITION 3.4. *The* **language model for Wikipedia category** *c_{w_i}, smoothed with the entire Wikipedia as a background corpus W, is the probability distribution over words (or phrases) in the collection of all Wikipedia articles $A_{c_{w_i}}$ under the category c_{w_i} and its subcategories:*

$$P_{c_{w_i}}(w) = (1 - \lambda)P(w|A_{c_{w_i}}) + \lambda P(w|W)$$

The **ranking of Wikipedia category** c_{w_i} *for a given debate* D *is in descending order of*

$$P(D|c_{w_i}) = \prod_{w_j \in D} P_{c_{w_i}}(w_j)$$

Note that these categories not only have the canonical facet name as a child, but may themselves be in a parent-child (or ancestor-descendant) relationship with each other. We retain these relationships as well, and later use them for the DAG structure of our **final topic graph**.

At this point we could directly use the hierarchy of the selected Wikipedia categories (perhaps, with heuristics to remove occasionally occurring cycles). However, this would yield a fairly noisy topic structure, as Wikipedia often exhibits unsystematic diversity (by its grassroots contributors) and does not enforce terminological standards (not to speak of ontological structures). For example, topics like "radioactive waste", and "nuclear safety" are part of a larger debate on "renewable energies", but Wikipedia does not organize them in this manner. Therefore, and inorder to build our **final topic graph**, we devises a clustering method on the Wikipedia categories, with preservation of whatever parent-child relationships are already present among categories.

Graph Coarsening Algorithm: To this end, we adopted and extended a *graph coarsening* algorithm, originally developed for the different task of multi-level graph partitioning [11]. We construct a weighted category graph as follows. For topology, we use the parent-child structure between Wikipedia categories.

DEFINITION 3.5. *We construct a node- and edge-weighted directed* **initial topic graph** $G_I = (V, E)$ *as follows. Each* $v_i \in V$ *corresponds to a wikipedia category produced by the previously described method. Let the category corresponding to* v_i *be denoted by* $c(v_i)$. $E = \{(v_i, v_j)|c(v_i)$ *is a parent of* $c(v_j)\}$.
The node weight *of* v_i*, denoted by* $w(v_i)$ *is the number of distinct facets that were mapped to* $c(v_i)$ *or one of its descendants.*
The edge weight for an edge $e_{ij} = (v_i, v_j)$*, denoted* $w(e_{ij})$ *is proportional to the number of common Wikipedia articles under* $c(v_i)$ *and* $c(v_j)$ *(and their subcategories). Let* $A_{c(i)}$ *and* $A_{c(j)}$ *denote the Wikipedia articles under* $c(v_i)$ *(and its subcategories) and* $c(v_j)$ *(and its subcategories), then* $w(e_{ij}) = \frac{|A_{c(i)} \cap A_{c(j)}|}{|A_{c(i)} \cup A_{c(j)}|}$*, the Jaccard co-efficient.*

So a category is considered more important if it transitively contains a large number of facets. We prune out very generic categories of portal nature and also very sparse categories of exotic specificity, by using upper and lower bounds α and β as thresholds for the category node weights.

From the initial topic graph constructed as described previously, we induce the final topic graph defined as follows.

DEFINITION 3.6. *A node- and edge-weighted DAG* **final topic graph** $G_F = (V, E)$ *is a hypergraph constructed such that each* $v_i \in V$ *corresponds to a topic denoted by* $t(v_i)$. $E = \{(v_i, v_j)|t(v_i)$ *is a parent of* $t(v_j)\}$*. Each topic* $t(v_i)$ *represents a graph* $G_S = (V', E')$ *such that* G_S *is a subgraph of the initial topic graph* G_I *(*$G_S \subset G_I$*) and* G_S *is induced by* V' *(*$G_S = G_I[V']$*).* V' *corresponds to a set of highly correlated Wikipedia categories.*

To construct the final topic graph, we run the graph coarsening algorithm of [11] on the initial topic graph, by gradually collapsing vertices and incident edges. In our experiments, we ran the algorithm until it arrived at around 500 clusters, and we generated a label (topic) for each cluster based on the most frequent n-gram in its underlying set of facet strings.

The graph coarsening algorithm works only with undirected graphs, it ignores edge orientation. So now we need to impose a DAG structure among the final clusters. This is done in two steps.

DAG Construction Phase 1: First, we aggregate the parent-child relations for two clusters A and B from their included categories $a \in A$, $b \in B$. For a being a parent of b in the original Wikipedia hierarchy, we compute the **average** of their edge weights into an aggregated weight $w(A \sqsupset B)$. For a being a child of b we analogously compute the aggregated $w(A \sqsubset B)$. Now the idea is to construct a parent-child edge from A to B if $w(A \sqsupset B) > 0$ and $w(A \sqsubset B) = 0$, and analogously for child-parent edges. Unfortunately, we cannot guarantee such conditions; we may have clusters with contradictory edges among their included categories.

DAG Construction Phase 2: As a second step, we, therefore, construct a priority order among all cluster pairs and then proceed in a greedy manner. We sort cluster pairs (A, B) in descending order of $priority(A, B) = w(A \sqsupset B) - w(A \sqsubset B)$. Now we construct parent-child edges between clusters in this order as long as $priority(A, B) > 0$. While doing this, we check for possible formation of cycles, and drop edges accordingly. Subsequently, we repeat this procedure for the dual priority order $w(A \sqsubset B) - w(A \sqsupset B)$ if it leads to additional edges without creating cycles. The greedy heuristic in this approach helps to ensure that the most prominent relationships between clusters are captured.

The outlined algorithm is our new way of constructing a topic DAG from Wikipedia categories, and completely mapping all collected facets onto these topics.

4. OPINION HOLDERS

We need to identify opinion holders who have expressed an opinion on these facets. For example, if we have extracted the facet "military action in Libya", then we need to find all people who support or oppose this particular facet. The problem here lies in reliably identifying, in the facet's surrounding text, the named entity who has expressed the opinion.

In order to identify these opinion holders, we again make use of the result snippets collected for extracting facets. Even though our seeds were of the form "he supports" (and variations thereof), it is often the case that the same sentence also names the opinion holder. For example, "Sarkozy said he supports military action in Libya" contains both the seed "he supports" as well as the opinion holder "Sarkozy". Our algorithm proceeds as follows:

1. Candidate identification: We run the Stanford dependency parser on the snippet's sentence that contains the support/oppose pattern, in order to identify the subject of the sentence.

2. Names-to-Entities mapping: In order to ensure that the subject is indeed a named entity, we made use of the YAGO ontology [18] which comprises all individual entities in Wikipedia and additionally provides a "means" relation which maps variations of names to the correct entities. For example, "B. Obama" and "President Obama" would both map to the entity Barack_Obama through the means relation. The means relation itself is constructed from Wikipedia cues like redirects and href anchor texts [9].

3. Disambiguation: As many names are ambiguous, YAGO actually connects the surface names to *all possible* meanings. For example, for "Obama"it provides both Barack_Obama as well as Michelle_Obama as entity candidates. Fortunately, YAGO comes with a simple but powerful heuristics for the *preferred meaning* of a name: the entity which most frequently occurs in Wikipedia as a link target for an href anchor text with the given name.

Using YAGO to identify names has the added advantage of canonicalizing the names of the opinion holders. In the case that YAGO does not know a name at all, then we discard the snippet. The disambiguation heuristics mentioned above may look crude, but it works amazingly well for important stakeholders like politicans or organizations. More advanced methods for named entity disambiguation can be easily plugged into our architecture.

We need also to consider the case when the parsing itself fails to identify a subject (e.g., because the subject is a pronoun or the snippet is not a grammatically correct sentence). In this case, we use the heuristics that the noun phrase identified by YAGO, which is closest to the seed is the opinion holder. This noun phrase may occur in the same sentence as the seed or in a previous sentence.

Finally, a candidate opinion is formed by adding the positively identified opinion holder, support or oppose (which is determined by the seed present in the sentence) and the facet. In addition, the snippet used to extract this opinion is added as its context.

5. SUPPORT/OPPOSE LEXICON

Using our initial seed set, we were able to extract ca. 7,000 opinion triples. However, the number of triples are limited by the initial seed set, since only those seeds are queried. For example, we find "she supports", but we would miss "she strongly supports" or "she is in favor of". Therefore, our next challenge is to *automatically* identify a large number of support and oppose patterns and create a lexicon of phrases. These phrases can then be used to collect more snippets from the Web and extract opinions from them. In this section we present several methods that we developed to automatically create a larger lexicon.

5.1 Unsupervised Method

Our first approach is an unsupervised method, bootstrapped from the initial seed set. We identify 10 support triples and 10 oppose triples from our initial list. From each triple, we isolate the person and the facet. Here we use the raw facets and entity names (not the canonicalized ones) because these are close to the variety of expressions used in news sources. Each ⟨person,facet⟩ pair is now viewed as a query submitted to a search engine. Examples of the resulting snippets are shown in Table 3. The bold text indicates phrases which could be used to populate the lexicon of support phrases. Our algorithm then processes these snippets in three steps:

1. We isolate the text in between the person and the facet phrase (e.g., "strongly defends US", "defended his decision to launch", etc). These substrings are now candidates for the lexicon, but still contain too much noise.

2. Therefore, we generate all n-grams (for $n = 1$ to 5) for the collected substrings, and each n-gram becomes a candidate.

3. Finally, the n-grams are ranked by their occurrence frequency in the collected snippets, and we retain the top-k.

Examples of collected phrases are shown in Table 4. We present an evaluation of their accuracy in Section 7.2.

Obama **strongly defends** US Military Action in Libya
Barack Obama has aligned himself with Nicolas Sarkozy by **committing to support** military action in Libya
US President Barack Obama **defended his decision to** launch military action in Libya
President Obama said Saturday that he **had authorized** limited military action in Libya
President Obama tells Americans that he **ordered** military action in Libya

Table 3: Examples of extracted snippets for (person,facet) pairs

support	approve, acknowledged, sponsored, campaigned for, strongly defends, had authorized, agree on
oppose	disapprove, campaigned against, criticize, overhaul, stand/speak against, to amend, attacks

Table 4: Examples of new seeds in the lexicon

5.2 Supervised Methods

We found that the unsupervised approach, despite its simplicity, is amazingly effective (see Section 7.2). Nonetheless, its recall and precision are limited. Moreover, we made the assumption that, if a query was formed by a "support" triple, then all snippets returned were candidates for the support set. However, the same ⟨person,facet⟩ query formulated from a support triple could also return an "oppose" snippet. To overcome these limitations, we developed methods based on supervised learning with different feature sets. These methods can be used to automaticly classify newly seen opinions.

Features. We explored the following features as inputs to the different supervised approaches.

1. NGRAMS: The NGRAMS feature for a snippet consists of all the n-grams extracted from its corresponding substring.

2. HMI: Hits Mutual Information: Consider an n-gram N (a candidate for a support or oppose phrase) and a *known support pair* s and a *known oppose pair* o. The number of times N co-occurs with s, and the number of times it co-occurs with o are indicators of whether the n-gram is a support or an oppose phrase. We formalize this approach into a measure HMI (derived from search-engine hits) as follows.
Let $S = \{s_i\}$ be the set of known support pairs and let $O = \{o_i\}$ be the set of known oppose pairs. Then,

$$HMI(N) = \frac{MI(S,N) - MI(O,N)}{MI(S,N) + MI(O,N)}$$

where $MI(S,N)$ is the mutual information of S and N, and computed as follows.

$$MI(S,N) = \sum_i P(s_i,N) \log \frac{P(s_i,N)}{P(s_i]P(N)}$$

where $P(s_i,N)$ is the probability that s_i and N co-occur and estimated as follows.

$$P(s_i,N) = \frac{hits(s_i,N)}{hits(s_i)hits(N)}$$

where $hits(s_i,N)$ is the number of hits returned by the search engine for a query formulated as "s_i AND N". If $HMI(N)$ is positive, it is more likely that N is a support phrase, otherwise, it is an oppose phrase.

3. SYN/ANT: Synonyms and Antonyms:

We add synonyms (SYN) to each verbal n-gram in the feature set (that is, any n-gram denoting a verbal phrase). Similarly, by substituting antonyms (ANT) in the verbal n-grams, we enhance the classifier in its ability to learn the opposite opinion. For example, for the n-gram "takes steps to change", we identify verbs "takes" and "to change". Adding synonyms to "to change", we arrive at a feature set that includes "takes steps to alter", "takes steps to modify", etc. Adding antonyms of "to change", gives us "takes steps to continue".

Classifiers. We trained two classifiers using different combinations of the features described above: a linear **SVM** and a J48 Decision Tree **(DT)**. In addition, we developed a nearest-neighbor-style classifier based on a statistical language model **(LM)** (similar to

[2]). In this approach, we construct LMs for both the support and the oppose n-grams as follows.

DEFINITION 5.1. *Let* $n^+ \in S$ *be a* **support n-gram**.*The* **LM for** n^+*, using* n^+ *itself and the set* S *for smoothing, is defined as the following probability distribution over n-grams:*

$$P_{n^+}(p) = (1-\lambda)P(p|n^+) + \lambda P(p|S)$$

where p *is an n-gram,* $P_{n^+}(p)$ *is the estimated probability of the n-gram in the LM of* n^+*,* $P(p|n^+)$ *is the probability of p in* n^+ *and* $P(p|S)$ *is the probability of p in* S*.*

Analogously, let $n^- \in O$ *be an* **oppose n-gram**. *The* **LM for** n^-*, using* n^- *itself and the set O for smoothing, is defined as:*

$$P_{n^-}(p) = (1-\lambda)P(p|n^-) + \lambda P(p|O)$$

Both $P(p|n^+)$*, and* $P(p|n^-)$ *are estimated based on the Jaccard similarity between* p *and* n^+ *or* n^-*, respectively. While* $P(p|S)$*, and* $P(p|O)$ *are estimated based on the max of the Jaccard similarities between* p *and each n-gram in* S *or* O*, respectively.*

Let Q be a newly seen n-gram. We now construct two queries as follows. A *synonym query* $Q_s = \{Q\} \cup \{Q_{SYN}\}$ and an *antonym query* $Q_a = \{Q_{ANT}\}$, where Q_{SYN} and Q_{ANT} are constructed using the SYN/ANT technique described previously. For the synonym query Q_s, the probabilities of generating it given either the support LM or the oppose LM are,

$$P(Q_s|n_i^+) = \prod_{p_j \in Q_s} P_{n_i^+}(p_j)$$

$$P(Q_s|n_i^-) = \prod_{p_j \in Q_s} P_{n_i^-}(p_j)$$

Similarly, we compute the probabilities of the antonym query Q_a. The newly seen n-gram is then classified based on the highest of these computed probabilities. (e.g. Q is classified as a support n-gram if $P(Q_s|n_i^+)$ is the highest).

Our prototype system for OpinioNetIt supports all the above described supervised and unsupervised approaches.

6. SYSTEM IMPLEMENTATION

We have implemented OpinioNetIt in Java. Our system consists of three main components as numbered in Figure 1.

1. Acquisition & Organization of Topics: (a) uses the dependency parser of the Stanford NLP package (SNLP) to extract opinions facets from the collected snippets, (b) makes use of available online resources (e.g. Wikipedia, Debatepedia) and maps each facet to canonical name, (c) constructs the controversial topics hierarchy, making use of online resources (e.g. Wikipedia, Debatepedia), and using different algorithms (e.g. coarsening algorithm of Metis [11]).

2. Acquiring Opinion Holders: (a) uses (SNLP) and its NER to extract candidate opinions holders from the snippets, (b) uses YAGO to map each candidate opinion holder to a canonical name.

3. Building the Support/Oppose Lexicon: (a) takes as input, a query formulated from an opinion holder and a topic. Using Google, it collects a set of relevant snippets, (b) uses (SNLP) to extract candidate seeds, (c) uses different classification packages (e.g. SVM-light, Weka) to classify the candidate seeds into support or oppose. All outputs are saved in a relational database. Opinions are saved in the RDF+text format, where the text consists of the context from which an opinion triple was extracted.

7. EVALUATION

For evaluating the accuracy of our automated methods, we performed a series of experiments. We first present end-to-end studies of the output quality of OpinioNetIt (Subsection 7.1). Then we present results on the most important components, namely, the construction of the topic DAG and the classification into support vs. oppose, with comparison to alternative methods (Subsection 7.2).

7.1 End-to-End Experiments

Data. A lexicon of 17 support/oppose phrases, shown in Table 4, served as seeds to extract ca. 30,000 opinion statements. Our sources Al-Jazeera (**ALJ**), BBC, CNN, New York Times (**NY**), Wikipedia (**WP**), and Google News (**GN**). Table 5 shows the breakdown of the opinions.

We restricted the number of snippets collected to the first 2,000 search engine results for each seed. These snippets yielded ca. 14,000 named entities, 23,000 facets, 4,000 canonicalized facets mapped to 5,000 Wikipedia categories, and finally 500 controversial topics organized in a DAG with ca. 3,000 topic-subtopic edges.

Methodology. We performed the following experiments for which we present precision results:

1. A *random sample* of 2,000 opinion statements (topic-politician pairs). A total of 18 judges assessed these opinions as "correct" or "wrong". For opinions that the judges deemed wrong, they could indicate whether: i) named entity is wrong, ii) facet is wrong, iii) relation (support/oppose) is wrong.
2. A small *focused sample* of 50 opinion statements by prominent politicians on important contemporary topics.
3. We compared our opinion network to a *limited ground-truth set* on prominent US politicians and their opinions about major topics, available on `procon.org` and `ontheissues.org`.

Metrics. The main measure of interest is the precision of the opinion statements, the fraction of truly correct outputs that our methods yield. We estimate precision by samples, with Wilson confidence intervals for statistical significance [20]. It is not obvious how to define and estimate a notion of recall (and F1 score) here, without manually reading the entire corpus. Thus we restrict this aspect of our studies to giving absolute numbers of different outputs obtained by our methods.

1. Results for Random Sample. Table 5 shows precision values for each source, and the overall precision. For the entire set of 2,000 samples, the micro-averaged precision was 72.4%. Given the difficulty of this extraction task, these results are quite satisfying.

Of all the opinions that were judged wrong, only 9% were due to the support/oppose relation. We conclude that our process of constructing the support and oppose lexicon is reasonably robust. The extraction of named entities and facets both contributed ca. 45% of the cases of incorrect opinions.

For named entity extraction, we used a heuristics in cases where the subject cannot be identified by the parser, which considers as an opinion holder the closest person name identified by YAGO. We found that this heuristic is not very robust. For example, it failed for the sentence "Prime Minister David Cameron met *Mr Gates* in Downing Street on Monday and restated *his support* for *American strategy in Afghanistan*".

Some of the incorrectly judged facets were very generic (e.g. "*George Clooney* on why *he supports the protests*", while the others were in long sentences which are difficult to parse (e.g. "*Ivan*

Source	Overall	Support	Oppose	ALJ	BBC	CNN	GN	NY	WP
#Opinions	29648	16011	13637	970	3349	2650	6861	9877	5941
Precision	0.724	0.7	0.75	0.72	0.66	0.78	0.788	0.74	0.69

Table 5: Opinions & precision results grouped by source

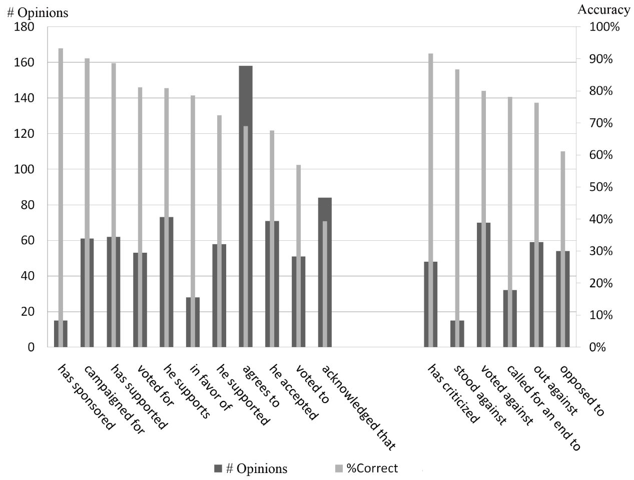

Figure 3: Accuracy of seeds.

Rojas said in Colombia that *he supports* Chavez's plan to secure *the release of his sister*").

The result that support opinions had lower accuracy than oppose opinions, is mostly due to difficult sentences with the pattern "acknowledged that" (e.g. "Kerry *acknowledged that* the bill would raise energy prices")

We notice significantly varying precision for different sources, with lower values for BBC and Wikipedia. For BBC, we encountered many snippets from user blogs with a different writing style compared to news. Wikipedia articles are long and detailed, and contain references to many different people. This writing style resulted in a lower accuracy in the extraction of the named entities.

2. Results for Focused Sample. We created a focused sample by manually identifying hot topics in each of the geo-political regions Africa, Asia, Europe, Middle East, and USA, and combining them with prominent politicians from these areas. An example subset of these pairs is shown in Table 6. For each cell of the table, we list the numbers of support ($+$) and oppose ($-$) opinions found in our corpus, and give the precision ($p = 0.93$) based on manual assessment by one of our judges.

	Africa: election in Cote d'Ivoire	**Asia: nuclear power plants**
Barack Obama	$2+, p = 1.0$	$2+, 2-, p = 1.0$
Laurent Gbagbo	$2-, p = 1.0$	
Binyamin Netanyahu		$1-, p = 0.0$
Nicolas Sarkozy	$2+, p = 1.0$	$1+, p = 1.0$
Manmohan Singh		$1+, p = 1.0$

Table 6: An example subset of the focused sample

3. Results for Ground-Truth Set. As the previous sample-based evaluation does not allow us to make any recall estimates, we also compared a subset of our automatically compiled opinions to a ground-truth dataset with largely aggregated and thus quasi-objective opinion polarities. To this end, we obtained the data on 8 US politicians (from both democratic and republican parties, e.g., Obama, McCain, etc.) on 19 controversial topics, from the two web sites `procon.org` and `ontheissues.org`. The topics were major themes such as abortion, same-sex marriage, death penalty, etc.

We aligned opinions triples in our database with ground-truth triples by first matching the named entities and then matching the ground-truth topics against i) the facets alone, ii) the Wikipedia topics of the facets, and iii) the Debatepedia topics of the facets. In the cases ii) and iii), we select the two topics with the highest Jaccard similarity to the ground-truth topic, based on the terms in the topics' facet strings.

Table 7 shows the results for precision and recall of this evaluation study. Note that the recall for facets is lower than for the Wikipedia or Debatepedia topics. This is caused by the inherent difficulty of matching fine-grained facets against coarse-grained main topics. For this situation, a recall of 30 percent is a satisfactory result.

	Facets	Debatepedia	Wikipedia
Precision	**0.98**	**0.81**	**0.80**
Recall	**0.30**	**0.42**	**0.41**

Table 7: Results for experiment with objective ground truth

We further investigated the reason behind the very high precision of both the focused set and the ground-truth set compared to the precision of the random set. Attributed to the nature of the topics in these sets, 96% of the matched triples were retrieved using the patterns: "support", "against', "voted for", "voted against". Sentences including these patterns are parsed with high accuracy. On the otherhand, different patterns with different parsing accuracies caused the low overall precision of the random set.

7.2 Components Experiments

1. Construction of Topic Hierarchy. We were interested in comparing our elaborate method for mapping facets to topics and for constructing the topic DAG against other alternatives like standard clustering. We presented judges with a randomly selected set of 500 facets along with: the top-3 canonical facets of each facet computed by the method of Subsection 3.2 (**LM-DebRank**), the corresponding top-3 Wikipedia categories of each canonical facet computed by the method of Subsection 3.3 (**LM-CatRank**), and the corresponding controversial topic of each category computed by the coarsening algorithm (**CO**) of Subsection 3.3).

Judges were asked to choose relevant: i) canonical facets for raw facets, ii) Wikipedia categories for canonical facets, iii) topics for Wikipedia categories. The 500 facets have 413 canonicalized facets, 387 Wikipedia categories, and 166 topics in the final topic DAG. In addition, the judges were asked to assess the correctness of 276 topic parent-child pairs from the topics in the final topic DAG. These topic pairs were collected using our DAG construction algorithm (**DAG**) described in Subsection 3.3. The main measure of interest is the precision of the assessed mappings.

Table 8 shows the results. **69%** of the raw facets f were assigned correctly to at least one of the top-3 canonicalized facets F, and **78%** of the presented F were correctly assigned to at least one of the top-3 Wikipedia categories C. Wrong (f, F) pairs were found mainly for very general facets (e.g., "the bill"). In such cases, informative LMs for facets require longer contexts.

For the (C,T) pairs obtained by the **(CO)** method, **73%** were correct, and **66%** of the facets have at least one correct complete path ($f \rightarrow F \rightarrow C \rightarrow T$). For the (T, T) pairs in the final topic DAG, **69%** were found to be correct, using the (**DAG**) approach.

As an alternative approach to both the (**CO**) and (**DAG**), we ran a hierarchical agglomerative clustering **(HAC)** algorithm on the collected Wikipedia categories. About **68%** of the 387 random (C, T) pairs were correct (**HAC-CT** vs. **CO**), while the 276 random (T, T) pairs have **71%** accuracy (**HAC-TT** vs. **DAG**). Since the construction of topics hierarchy depends on the construction of each topic cluster, the precision of these two components togother is computed. The precision of **CO** and **DAG** is higher than the precision of **HAC-CT** and **HAC-TT**. Also when comparing the judiciously constructed DAG vs. standard HAC, one also needs to consider the overall size and structure of the resulting graphs. DAG produces a compact and nicely explorable hierarchy with a total of 533 nodes and 299 edges (so that most nodes are leaves in the DAG). In contrast, HAC creates a much larger binary tree with 1056 nodes and 1055 edges in total. This graph is much more tedious to explore, and does not convey the same level of informativeness as our constructed DAG.

Items(#)-Pairs (#)	**Approach**	**Precision**
f **(500)-(**f**,**F**) (1500)**	LM-DebRank	0.69
F **(413)-(**F**,**C**) (1239)**	LM-CatRank	0.78
C **(387)-(**C**,**T**) (387)**	CO	0.73
nodes (137)-(T**,**T**) (276)**	DAG	0.69
–	CO&DAG	0.50
C **(387)-(**C**,**T**) (387)**	HAC-CT	0.68
nodes (532)-(T**,**T**) (276)**	HAC-TT	0.71
–	HAC-CT&HAC-TT	0.48

Table 8: Results for controversial topics graph

2. Support/Oppose Classifiers. For the unsupervised approach described in Section 5.1, our judges evaluated 460 n-grams out of 2,000 support or oppose n-grams. For the supervised approaches, 540 n-grams were manually labeled as support or oppose. Out of these, 400 n-grams for training, and 140 n-grams for testing different approaches (**SVM, DT, LM**) using different combinations of features (**NGRAMS, HMI, SYN, ANT**).

The main measures for the unsupervised approach are relative precision and relative recall (based on the 460 samples), while precision and recall are used for the supervised approaches (based on the withheld 140 test cases).

Table 9 shows the best results for the different methods using different combinations of features. The unsupervised approach outperforms the other methods, despite its simplicity. The LM method performs better than DT and SVM. Using n-grams only as features gives bad results because of data sparseness. For this reason, we used aggregated features (e.g., sum, max) over synonyms and antonyms features. Using synonyms only in the similarity measures gives better results than using antonyms as well while results with sum outperform the results of max function.

Approach	**Precision**	**Recall**
Unsupervised	**0.79**	**0.60**
SVM+Ngrams+HMI	0.47	0.21
SVM+SYN+SUM	0.61	0.60
DT+SYN+SUM	**0.63**	**0.62**
LM+SYN+SUM	**0.69**	**0.70**

Table 9: Results for support/oppose classifiers

8. USE CASES

As use cases for knowledge discovery and political analyses, we briefly discuss scenarios on the phenomena of so-called flip-flops and dissenters.

Flip-Flops Detection. A flip-flop according to Wikipedia is a sudden real or apparent change of policy or opinion by a public official. To detect flip-flops, we group opinions collected by our system by pairs of politicians and topics, both in canonicalized form. If a group contains both support and oppose opinions, we consider the politician to have flip-flopped. Table 10 shows some examples that we found in our collection (O denotes the opinion polarity: "+" for support, "-" for oppose).

Dissenters Detection. Our goal here is to detect interesting deviations from correlated opinions on different topics. Often, once you know that a person supports topic A, it is clear that she or he also supports topic B, given the semantic connection between A and B. To identify such topic pairs, we compute correlation coefficients and select the most positively correlated pairs. Then we aim to find notable dissenters: people who deviate from this pattern and support only one of A or B and oppose the other one. Table 11 shows some interesting examples that we found in our collection (dominant opinions: "++", "+-", "-+", or "–" with "+" denoting support and "-" denoting oppose).

9. RELATED WORK

Opinion mining (aka. sentiment mining) is a richly researched topic that gained a lot of attention in recent years for business intelligence, smart advertisements, marketing campaigns, etc. The standard model for these mining tasks is to identify objects (e.g., cameras or movies), facets (aspects, features) of these objects on which a sentiment is expressed (e.g., the camera's ease of use, or the movie's special effects), and a polarity of the opinionated statement (positive or negative), and sometimes also an opinion strength for quantifying the polarity. Typically, these cues are aggregated to form opinion profiles, based on many opinions like customer reviews or postings in discussion forums. [16, 13] are excellent surveys on the methodologies for doing this kind of opinion analysis. Our work uses specific techniques from this state of the art, but adapts them to our setting and combines them with new methods. Moreover, the opinion holder is crucial to our task, discriminating our goal from that of traditional sentiment mining: we are interested in the individual entities that express opinions rather than merely aggregating over many individuals.

Analyzing political opinions on controversial topics is inherently more difficult than standard sentiment mining (for products, movies, etc.), given the complexity of the topics and the subtleties in expressing opinions on them. For example, the SentiWordNet lexicon [7], although very useful for understanding product reviews, turned out to be of little help on political texts as these have much richer phrases, as opposed to polarity-bearing single words like adjectives and adverbs. Thus, the theme of political opinion mining is still in its infancy. The little prior work that we are aware of is scattered across specific sub-issues. There is some work on classifying political texts with regard to political parties [19, 15, 21], most notably, democrats vs. republicans in the US, but this stays at the crude level of general attitude and does not address positions on specific controversies. This theme of coarse-grained classification is taken further by the work of [10], which aims to annotate political speeches and parliament debates, but does not deal with fine-grained topics. Work on question answering has also tapped into opinion-related questions (e.g., [3]), but only for providing aggregated answers, not for constructing systematic opinion networks. [14] uses semantic taxonomies to identify aspects of topics, and analyzes opinions on these aspects rather than objects as whole. This applies to sentiment mining on politicians (with aspects such as Vietnam war, Watergate affair, etc.), but it does not address the more far-reaching

Politician	Topic	O	Example Snippet
Barack Obama	nuclear power	+	... Obama says he supports nuclear power ...
		−	... Obama criticized his rival McCain's proposal to encourage the building of 45 new nuclear reactors ...
Angela Merkel	nuclear power	+	... Dr. Merkel's government support for the nuclear energy sector ...
		−	... Chancellor Angela Merkel said she aimed to accelerate Germany's move away from nuclear energy ...

Table 10: Examples of flip-flops detected by our system

Topic Pair	Dominant $(+-)$, $(-+)$	Dissenter $(--)$, $(++)$
Emissions Trading & Offshore Drilling	⟨Cynthia McKinney⟩ ⟨+⟩ ⟨Carbon Neutrality⟩ ⟨Cynthia McKinney⟩ ⟨-⟩ ⟨Offshore drilling⟩	⟨John McCain⟩ ⟨+⟩ ⟨Emissions trading⟩ ⟨John McCain⟩ ⟨+⟩ ⟨Offshore drilling⟩
George W. Bush Politics & Abortion	⟨Sarah Palin⟩ ⟨+⟩ ⟨George W. Bush⟩ ⟨Sarah Palin⟩ ⟨-⟩ ⟨Abortion⟩	⟨Joe Biden⟩ ⟨+⟩ ⟨George W. Bush⟩ ⟨Joe Biden⟩ ⟨+⟩ ⟨Abortion rights⟩

Table 11: Examples of topics correlations and dissenters

goal of connecting opinion holders with fine-grained controversial topics. Finally, the work described in [2] language models to classify pro/con statements, but disregarded the opinion holders.

For gathering statements that connect individual people (mostly politicians) with opinionated expressions, we make use of techniques for information extraction from text and Web sources. [5, 17] give overviews of state-of-the-art methods. Our approach builds specifically on the pattern-fact duality principle that goes back to [4, 1] and allows us to bootstrap the acquisition process with very few seeds. We customize this general framework to collect opinionated cues rather than general relational patterns. For matching names in text sources against canonical entities, we leverage existing knowledge bases like DBpedia, Freebase, or Yago. We specifically make use of the *means* relation that Yago [18] provides for individual entities and their lexical name variations. This information is in turn derived from anchor texts and redirects of Wikipedia. All these are building blocks of our methodology for constructing OpinioNetIt, but each ingredient has an auxiliary role only.

10. CONCLUSIONS AND FUTURE WORK

In this paper, we described OpinioNetIt, an opinion-base of facets, opinion holders and their opinions. First our system acquired opinions and extracted facets from Web result snippets using an initial set of seed patterns. These facets were then canonicalized and hierarchically organized. Second, for each facet, an opinion holder was identified from the same snippets. Third, using a small set of facts, a lexicon of support/oppose phrases was constructed. Finally, new patterns from the lexicon were used to collect further facts. Our evaluation showed an overall precision of 72%.

As future work, generic facets (such as "the bill", etc.) extracted for some opinions require a closer look at the snippet, and possibly the article itself. Also and in order to improve the accuracy of our system, we either need to predict an inaccurate extraction or use additional heuristics in addition to the deep parsing techniques which are sometimes limited when it comes to long sentences. Finally, we plan to expand our techniques to other kinds of sources such as blogs and online forums.

11. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *ICDL*, 2000.

[2] R. Awadallah, M. Ramanath, and G. Weikum. Language-model-based pro/con classification of political text. In *SIGIR*, 2010.

[3] A. Balahur, E. Boldrini A. Montoyo, and P. Martínez-Barco. A unified proposal for factoid and opinionated question answering. In *COLING*, 2010.

[4] S. Brin. Extracting patterns and relations from the world wide web. In *Selected papers from the Intl. Workshop on The World Wide Web and Databases*, 1999.

[5] A. Doan, L. Gravano, R. Ramakrishnan, and S. Vaithyanathan. Special issue on managing information extraction. In *ACM SIGMOD Record*, 2008.

[6] S. Elbassuoni, M. Ramanath, R. Schenkel, and G. Weikum. Searching RDF graphs with SPARQL and keywords. In *IEEE Data Engg. Bulletin*, 2010.

[7] A. Esuli, S. Baccianella, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC 2010*, 2010.

[8] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, Enschede, 2001.

[9] J. Hoffart, M. Amir, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP 2011*, 2011.

[10] R. Kaptein, M. Marx, and J. Kamps. Who said what to whom?: capturing the structure of debates. In *SIGIR*, 2009.

[11] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.*, 20(1), 1998.

[12] D. Klein and C. Manning. Accurate unlexicalized parsing. In *ACL*, 2003.

[13] B. Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*. 2010.

[14] Y. Lu, H. Duan, H. Wang, and C. Zhai. Exploiting structured ontology to organize scattered online opinions. In *COLING*, 2010.

[15] T. Mullen and R. Malouf. Taking sides: User classification for informal online political discourse. *Internet Research*, 18(2), 2008.

[16] B. Pang and L. Lee. Opinion mining and sentiment analysis. *FnT in IR*, 2(1-2), 2008.

[17] S. Sarawagi. Information extraction. *FnT in Databases*, 1(3), 2008.

[18] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW*, 2007.

[19] M. Thomas, B. Pang, and L. Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *EMNLP*, 2006.

[20] E. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158), 1927.

[21] B. Yu, S. Kaufmann, and D. Diermeier. Classifying party affiliation from political speech. *Information Technology & Politics*, 5(1), 2008.

[22] C. Zhai. Statistical language models for information retrieval. *FnT in IR*, 2(3), 2008.