Automatic argumentation mining and the role of stance and sentiment

Manfred Stede, University of Potsdam

PREPRINT – to appear in *Argumentation in Context*, 2020

Abstract

Argumentation mining is a subfield of Computational Linguistics that aims (primarily) at automatically finding arguments and their structural components in natural language text. We provide a short introduction to this field, intended for an audience with a limited computational background. After explaining the subtasks involved in this problem of deriving the structure of arguments, we describe two other applications that are popular in computational linguistics: sentiment analysis and stance detection. From the linguistic viewpoint, they concern the semantics of evaluation in language. In the final part of the paper, we briefly examine the roles that these two tasks play in argumentation mining, both in current practice, and in possible future systems.

Keywords: argumentation structure, argumentation mining, sentiment analysis, stance detection

1. Introduction

Early approaches to the automatic analysis of argumentation in fact date back to the previous century (e.g., Cohen 1987), but only about five years ago did the task attract wider attention, and a subfield of "argumentation mining" has been established in computational linguistics (CL). This is demonstrated for instance by a continuous series of international workshops co-located with important CL conferences; by an increasing number of related articles in both CL and argumentation journals; or by the establishment of various interesting R&D projects both in industry and in academia. Argumentation mining is not a homogeneous endeavour but rather a family of subtasks, some of which are relevant only for certain specific applications. Broadly, the field aims at finding the components of arguments in linguistic data, and identifying the relations between them (mostly: support and attack). Additional research tries to automatically detect the underlying argumentation schemes, to name implicit premises, or to assess the quality of arguments - all of which are considerably more difficult than the "base tasks" listed above. Further add-on tasks that build upon the mining steps are the automatic *summarization* of a set of arguments on the same topic, or the *production* of argumentative texts from components that have been gathered from text collections. The latter step then paves the way to systems that would be capable of interactive argumentative exchanges.

This paper aims at providing a brief overview of the goals and methods of automatic argumentation mining, intended for an audience whose background is primarily in argumentation studies rather than in computational linguistics or computer science;[1] furthermore, it discusses two other popular CL tasks - sentiment and stance analysis - and then turns to assessing the potential role of those tasks in argumentation mining. From the Linguistics perspective, the connection between the various realms is the language of *evaluation*: We will see that sentiment and stance address those aspects of evaluation that revolve around *polarity* or *valence* (i.e., positive/negative attitude), and that these aspects can be fruitfully exploited for argumentation mining, although the computational methods used in present-day implementations mostly provide only quite coarse-grained and often error-prone accounts of evaluative meaning.

For reasons of space, our playing field will be largely restricted to that of analysing monologue written language (conventional text, social media), while phenomena of dialogical exchanges are being excluded. Specifically, we will in various places use examples from the "argumentative microtext corpus" (Peldszus/Stede 2016) - a collection of short texts that have been collected to provide relatively simple material for bootstrapping approaches to the automatic analysis of argumentation, and to foster the in-depth study of linguistic phenomena of argumentation. As shown on the corpus website[2], the texts have been annotated not only with argumentation structure, but (in part by other researchers) also with representations of discourse structure, semantic features, argumentation schemes, and hidden premises.

We begin in Section 2 by briefly looking at applications that recent argumentation mining has targeted, as this angle sheds light on the various motivations for developing computational approaches. Then, Section 3 discusses some approaches to represent the structure of argumentation that is to be extracted from text, and Section 4 summarizes efforts on the various computational subtasks needed to achieve such structure inductions. Section 5 introduces sentiment and stance analysis, two older and quite popular tasks in CL, and then Section 6 analyses the potential connections between those two approaches to evaluative meaning on the one hand, and the problem of argumentation mining on the other. The paper ends with some conclusions in Section 7.

2. Some applications of argumentation mining in text

Early work on argumentation mining addressed legal documents, in particular court decisions, and aimed at labelling sentences with information on whether they express a conclusion drawn by the court, a statement supporting the conclusion, or one opposing it (Palau and Moens, 2009). Technically, in line with the majority of work up to today, the authors manually annotated a text corpus with these labels, then identified a set of observable linguistic features that can be automatically extracted from the text. (Examples of such features will be given in Section 3.) An automatic classifier builds a model of the complex correlations between those feature values and the target categories (i.e., conclusion, support, oppose), which can then be applied to categorize new units of text. Comparing the classifier output to the manually-assigned labels, the authors achieved an F-measure of 74% for detecting conclusions and 68% for the two

[1] A slightly more extensive survey from a technical perspective was presented by Lippi and Torroni (2016); and a somewhat more elaborate discussion of the field is provided by Stede and Schneider (2018).
[2] http://angcl.ling.uni-potsdam.de/resources/argmicro.html

types of premises. Palau and Moens also added a rule-based module (based on a context-free grammar) that constructed a tree structure linking the various argument components to each other.

Today, data-driven machine-learning approaches are clearly being favoured over analyses on the basis of hand-written rules, and thus many different corpora have been developed, which contain task-specific annotations. (For an overview, see Stede/Schneider 2018, ch, 4.) Very often, the data stems from social media or from web platforms that encourage users to exchange arguments, such as *createdebate.com*. On some platforms, the user contributions are already being sorted as to whether they are for or against the critical question that initiated the discussion. Text annotations range from premise/conclusion labels to full argument structure, such as in the web text corpus by Habernal and Gurevych (2017), which follows a variant of the Toulmin scheme (Toulmin 2008). Examples from other text genres are the corpus of newspaper editorials by Al-Khatib et al. (2016) or the persuasive student essay corpus by Stab and Gurevych (2014). The latter points to another application that some researchers are investigating: the contribution of argumentation mining to automatic essay scoring. The idea is to incorporate the presence or the complexity of argumentation in the student's essay into the process of automatically assigning it a grade (e.g., Persing/Ng 2015).

The challenges of retrieving arguments, on a topic to be set by the user, from the web are discussed by Wachsmuth et al. (2017a), and the prototype of such a search engine, which gathers material from debate web sites (as mentioned above) is available online.[3] A similar effort of making sense of noisy data is the work on analysing the argumentation in presidential debates by Lawrence and Reed (2017).
Finally, an even more ambitious application that presupposes somewhat more technology than the "mining" functionality that we sketched so far is the idea of an automatic debating system that can engage in a discussion – in typed or even in spoken language – with a human opponent. This setting is addressed by the IBM Debater project[4], which in 2018 reached the milestone of a first public demonstration. When being given the topic to be debated, such a debating system locates, analyses and possibly re-structures arguments that it finds in vast amounts of text. In addition, it needs to linearize that information into a coherent text, and attach a powerful speech synthesis component. (And, on top of this, the system should be able to understand and react to the contributions made by the human opponent.) In other words, today's automatic debaters are not in fact forming a viewpoint and generating language "from scratch" by their own reasoning but by harvesting material from huge text resources. Therefore, finding arguments in unstructured text, for instance on the web, remains a central task.

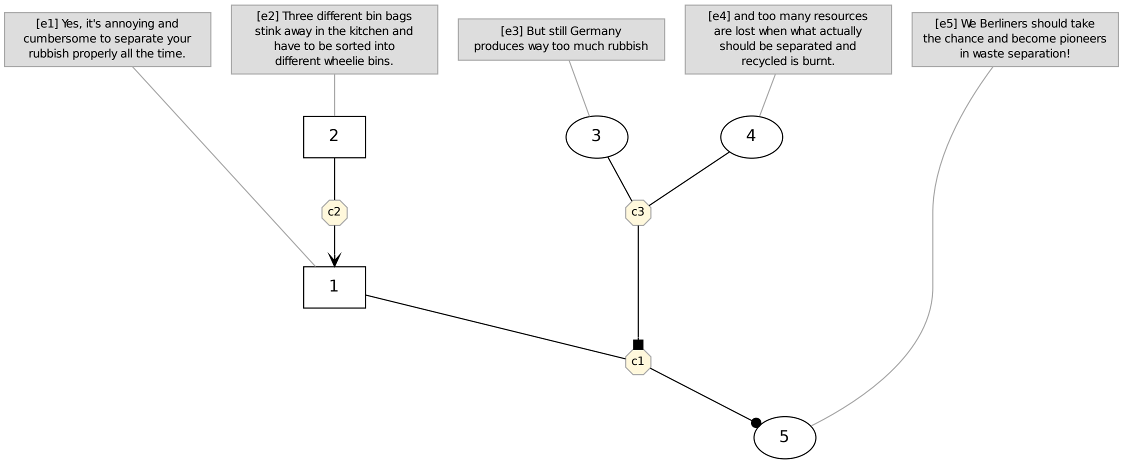
3. Representing the structure of argumentation

The desired output of an argumentation mining system obviously depends on the specific task that the system is designed to tackle. While for many purposes it is sufficient to identify claim and premise sentences, sometimes a "deeper" analysis is desired, which can account for nested structures, or distinguish different kinds of

[3] https://args.me
[4] https://www.research.ibm.com/artificial-intelligence/project-debater/

support/attack relations. To that end, as mentioned earlier, Habernal and Gurevych (2017) designed an annotation scheme that slightly modifies the approach of Toulmin (1958). Another scheme for annotating such structures, inspired by the work of Freeman (2011), is described by Peldszus and Stede (2013). (We will refer to this scheme in the subsequent sections of the paper.) The representation is a graph where units of text (sentences, clauses) form the nodes, and labelled arcs represent relations between them. (For illustration, see Figure 1 below.) Serial support is accounted for by establishing the relation not only between premise and claim, but also between premises. In the relation set, the scheme distinguishes two kinds of attack, viz. rebut (dispute the proposition or speech act corresponding to a node) and undercut (dispute the relevance of a purported premise for a conclusion). Thus emphasizing the dialectical nature of argumentation, the scheme can be employed to represent the argument made in a text that states and defends a certain claim, and possibly considers potential counter-arguments along the way. A corpus that has been annotated with these structures is the "argumentative microtext corpus" by Peldszus and Stede (2016) and Skeppstedt et al. (2018), mentioned in Section 1. The texts are deliberately kept short, so that they could effectively be annotated for different linguistic phenomena in addition to the argumentation structure, and hence correlations can be studied. Figure 1 shows a text from the corpus along with its argumentation structure, which happens to include both types of attack mentioned above. The guidelines for human annotators explain the criteria for deciding on segmenting the text into 'argumentative discourse units' and relating them to one another; they are available on the corpus website.

For larger texts, annotation of such structures becomes somewhat more complicated, but not fundamentally different. We mention here the scheme underlying the persuasive essay corpus compiled by Stab and Gurevych (2014). Geared toward the generic structure of student essays, for every paragraph of the essay a tree similar to the ones of the Peldszus/Stede scheme is being annotated. In addition, a text is supposed to have a 'major claim', located toward the beginning of the text, which is superordinate to the various paragraph-level claims.


[e1] Yes, it's annoying and cumbersome to separate your rubbish properly all the time.
[e2] Three different bin bags stink away in the kitchen and have to be sorted into different wheelie bins.
[e3] But still Germany produces way too much rubbish
[e4] and too many resources are lost when what actually should be separated and recycled is burnt.
[e5] We Berliners should take the chance and become pioneers in waste separation!
2
3
4
c2
c3
1
c1
5

Fig, 1: Sample text and analysis from the first part of the "argumentative microtext corpus" (Peldszus/Stede 2016). Circled nodes represent the viewpoint of the proponent (author), boxed nodes that of the imaginary opponent. Arrowheads denote support, disc-shaped heads denote rebut. Nodes 3 and 4 are marked as playing a joint role in undercutting (square head) the relation 1-5.

4. Argumentation mining: subtasks

With its range of diverse applications, argumentation mining is not a single and well-defined task but rather a family of related subproblems, some of which are to be selected when working toward one particular application. In the following, we briefly describe the most important subtasks.

4.1 Find claims

Given an opinionated text, what does it argue for? Being able to identify the central claim in a text is a fundamental aspect of argumentation mining. It can be carried out, for instance, in student essays or in newspaper editorials, i.e., in texts that are known to be argumentative. We mention here the work by Falakmasir et al. (2014), who did extensive experiments for finding "good" (predictive) features for identifying the thesis and the conclusion sentences in student essays.[5] The most useful ones were positional features (sentence number in paragraph, paragraph number in text); certain syntactic features (e.g., presence of prepositional and gerund phrases, number of adjectives and of adverbs); presence of words from a predefined list including *although, even though, because, due to, led to, caused*, and others; and finally essay-specific features such as the number of words in a sentence that overlap with the words in the essay prompt.

Sometimes it may be necessary to look for claims in less "orderly" text material. Recall for example the *Debater* application, where the system is given a controversial topic, and then it needs to mine huge amounts of text for arguments that are in favour or against, in order to prepare a debate speech and to be able to react to the opponent's speech. This setting differs from the one described above: Rather than identifying the one claim sentence in a given text (where the existence of that sentence, at least in the case of a student essay, can be safely assumed), we now sift through a large number of text documents, trying to find a sentence that is (a) on the target topic, and (b) qualifies as a claim. One interesting approach to this problem is described by Shnarch et al. (2017). Instead of leaving feature selection entirely to a "black box" classifier, as is customarily done, they devise a three-step procedure for learning a different kind of model.

First, a variety of analysis tools is applied to the sentence, and every word is turned into a vector that in addition to the word contains a number of attributes derived by the tools. We illustrate this vector for the example of the word *argue* occurring in some sentence (Schnarch et al. 2017, p. 1347): it consists of part-of-speech (`VERB`); its syntactic relation in the parse tree (`ROOT`); its hypernyms taken from *WordNet* (Fellbaum 1998) ({`present, state, express`}); its presence or absence in a task-specific claim lexicon (+), in a lexicon of 'sentiment words' (see Sct. 5) (−), and in a lexicon listing terms belonging to the topic under consideration (−).[6]

[5] The 'thesis' is the statement of the standpoint that the text argues for, and the 'conclusion' at the end of the text summarizes the main point; it can reiterate the thesis or add further information.

[6] The lexicons are derived from large corpora of claim sentences and text known to discuss a specific topic. We use + to denote presence in the lexicon, and − for absence.

Second, when a (labelled) training corpus of claims and non-claims has been "enriched" in this manner, they statistically select those features that significantly correlate with the category to be predicted (i.e., claim / non-claim). All the vectors are thus being reduced to the most useful dimensions.

Third, they use an algorithm that iteratively builds *patterns* of sequences of such vectors, where either an attribute is added to an element in a pattern, or a new element is being added to the right; again, these decisions are guided by computing their statistical utility for the classification task. In the end, the set of patterns represents declarative knowledge of "what is a claim", and is subject to inspection or modification by researchers. Shnarch et al. evaluated their system against various competing approaches, including a convolutional neural network as often used today for such tasks, and found that their system wins by a large margin, in the best configuration yielding a precision of 0.42 for finding claim sentences.

Finally, we point out that a text, even if it can be taken to be argumentative, need not have an explicit claim. This can be the case for instance in newspaper editorials, where the "point" can be made by giving supporting statements, and the claim may be left implicit. Also, the second part of the "microtext" corpus has a number of examples where the claim that the author had left implicit was then added by annotators as extra information (Skeppstedt et al. 2018).

A class of texts where implicitness abounds is exchanges on certain social media such as chats or Twitter. Wojatzki and Zesch (2016) studied tweets belonging to a debate on atheism, where users regularly make statements from which their position on the topic is to be inferred. One of the examples from the corpus is
Bible: infidels are going to hell!
And the analysis of the authors is that the user has expressed a positive stance toward Christianity by quoting from the Bible and by acknowledging the existence of hell, so that the tweet serves as a premise for the implicit claim *I am against atheism*.

4.2 Find premises

The minimum configuration of an argument is generally taken to be a claim and a supporting statement (premise), where the term 'premise' is sometimes used to also subsume opposing statements, which may well be part of an argumentation in a monologue text (cf. Section 3).

In analogy to the situation with claim detection described in the previous subsection, we can distinguish the two basic scenarios of (i) finding the premises in a text (or portion of a text) that is known to be argumentative – and where the claim is possibly already found – and (ii) finding premises in a large text collection, such that they support a claim that has been given by a user or by some other part of the software.

For (i), it can be assumed that premises are located in the neighbourhood of claims, and many approaches try to exploit the local text coherence for identifying premises: Supporting statements are often explicitly linked to the claim by connectives that signal a cause/reason relation, such as *because, since, therefore, thus*, and many more. Conversely, opposing statements typically need to be marked by concessive or contrastive connectives (or longer phrases) so that the reader perceives the shift of perspective from a *pro* to a *contra* view, and back again: “Even though the camera is expensive, you should buy it, because the sensor is absolutely state-of-the-art.” Obviously, claim, support and objection can be arranged in different linearizations, and the corresponding sequence of connectives needs to be detected. In addition to the relations just mentioned, one can also look for additive connectives, which may signal additional premises with the same function as the previous one: “Furthermore, the lens has exceptional quality.” Most argumentation mining systems employ connectives as features when learning models; one study that specifically focused on exploiting connectives of various kinds is that of Eckle-Kohler et al. (2015) for German (but the findings can easily be carried over to other languages).

Naturally, supporting statements can also be given without an explicit connective linking it to the context. Many models aim to learn large collections of words that tend to co-occur in the two text spans of specific semantic relations. Classical examples are *push* and *fall* for causal relations, and the numerous lexical antonyms (*large* – *small*, etc.) for contrast relations. Building up an inventory of such lexical knowledge would in principle require huge amounts of text labelled with the relations in question, if a reasonable coverage is to be achieved. Since annotated text in such quantity is generally not available, many researchers use the idea of extracting the lexical pairs from sentences that are linked by a suitable connective. For example, a construction *A, because B* is used to extract word pairs from *A* and *B*, and these pairs are assumed to also be indicative for sentences linked by a causal relation where no connective is present. One study that adapted this approach to the task of finding argumentative support relations is that of Biran and Rambow (2011), who used a corpus of annotated Wikipedia discussion pages for evaluating lexical knowledge acquired from the much larger, and unlabelled, general Wikipedia, in the way sketched above. Their system achieved an F-measure of up to 0.5 for finding claim and support sentences.

In scenario (ii), which is often called ‘open domain argument construction’, topic and claim are given, and evidence for supporting the claim are to be found in large text collections. This task seems not very different from that of claim identification, and thus the techniques used are usually quite similar. For example, Shnarch et al. (2017) applied their pattern-building approach (see Section 4.2) also to evidence identification. It turned out to be more difficult that claim detection, though, with F-measures running up to only 0.35.

4.3 Build a representation of argumentation structure

The problem of finding claim and premises in an argumentative text can be implemented as a straightforward ‘sequence labelling’ problem, where in a sequence of minimal units of analysis, each unit is assigned a label such as: `claim/support/attack/none`. For illustration, here is a text from the argumentative microtext corpus, segmented into ‘argumentative discourse units’ (ADUs for short):

Should there be a deposit on glass bottles?
[I live in Michigan, where we have a deposit.]$_1$ [I regularly see people bringing huge amounts of bottles and cans to grocery stores to recycle them.]$_2$ [While I do not have exact statistics on the topic,]$_3$ [this indicates to me that deposits promote recycling and should therefore be adopted.]$_4$ [This has also become a source of income for homeless

people.]$_5$ [I see homeless people picking up littered cans so they can later recycle them for the deposit, which seems good to me.]$_6$

The output of such a labelling module should be:

```
1: none / 2: support / 3: attack / 4: claim / 5: support / 6:
support
```

In the annotation of the corpus, however, which uses the scheme sketched in Section 3, segment 6 is marked to support 5, which in turn supports the claim 4. This is an instance of 'serial support', which cannot be inferred from the mere sequence of labelled units.

For some purposes, a 'flat' labelling of ADUs is sufficient, and the result then presupposes that every segment is implicitly related directly to the claim. If, on the other hand, one is interested in a more informative overall structure, then different relations between units need to be captured explicitly (including cases of recursion), and a more powerful representation along the lines described in Section 3 (and illustrated in Fig. 1) is needed.

One technical approach to solving this graph-building problem is to formulate it as a two-step process: First, apply classifiers to each segment, which determine probabilities for the possible roles that a segment can play in a text. Then, use an optimization procedure that accounts for the fact that a well-formed structure is to be produced overall, and aims at finding the most likely such structure, given the probabilities from step 1. One computational framework for implementing step 2 is *Integer Linear Programming (ILP)*, which allows the programmer to formulate well-formedness constraints, and then takes care of the numerical optimization task. Stab and Gurevych (2017) explain one solution along these lines, which produces argumentation structures for the persuasive essay corpus of Stab and Gurevych (2014). Another implementation is presented by Afantenos et al. (2018), who demonstrate that ILP performs more or less on a par with a second approach called the *evidence graph* model. Here, the well-formedness of the output structure in ensured by means of a standard algorithm from graph theory (*minimum-spanning tree, MST*). In a nutshell, one first constructs a fully-connected graph where all ADUs of the text are linked to all others; the results of the classifiers that try to inspect the function of the individual ADUs (step 1 above) are being mapped to probabilities and associated with the edges of that graph. Then, the MST algorithm extracts a subgraph such that it fulfils the condition of being a tree (there is a single root node, the claim; there are no cycles; every node has exactly one outgoing edge, except the claim, which has none) and that maximises the probabilities associated with the edges.

Not surprisingly, automatically building such trees that exactly correspond to the manually-annotated ones is quite difficult even for short texts such as the microtexts. While the results of Afantenos et al. for the subtasks in isolation are not bad (e.g., the F-measure for claim detection is 0.88), only few predicted trees are in complete correspondence to the manual annotations.

Finding claims and evidence, and building a structure representation can be considered the "core tasks" of argumentation mining. We point out, however, that a range of additional (and rather difficult) analyses are being studied in the field, such as the detection of underlying argumentation schemes (e.g., Feng/Hirst 2011), the explicitation of enthymemes (e.g., Boltuzic/Snajder 2016) and the assessment of the quality of arguments (e.g., Wachsmuth et al. 2017b).

5. Capturing evaluation: Automatic stance detection and sentiment analysis

Despite a lot of research over the past decades on computational semantics, ontologies, or large-scale resources of "world knowledge" on entities and event types, computers are so far not able to robustly grasp the content of a sentence or text, connect it systematically to prior knowledge, and draw inferences of the kind that humans do intuitively and with great ease. (Exceptions occur when the relevant knowledge and inference calculus is carefully modelled by hand for a small domain.) As became clear in the previous section, current argumentation mining is driven by surface features that can be computed from the text directly or with the help of a syntactic parser or similar tools; the *meaning* of the units involved is hardly being accounted for.

Notwithstanding this pessimistic observation, there are certain aspects of linguistic semantics that nowadays can be tackled automatically with some success. In the following, we briefly explain the computational tasks of *stance detection* and *sentiment analysis*; Section 6 will then discuss in what way they can support argumentation mining.

5.1 Stance detection

In the results compilation of a recent "shared task" (i.e., a competition among researchers working on the same dataset) in the CL community, stance detection is defined as "the task of automatically determining from text whether the author of the text is in favor of, against, or neutral towards a proposition or target. The target may be a person, an organization, a government policy, a movement, a product, etc. " (Mohammad et al. 2016, p. 31). Popular application domains for this problem are product reviews, where the product name is given to the system as target, and its job is to assign a label such as `favor/against/neutral` to a review. Many other domains have been exploited, though; for example, the aforementioned shared task used a dataset of tweets whose targets were atheism, climate change as a real concern, feminist movement, Hillary Clinton, and legalization of abortion. Hence, stance detection can be employed to sift through social media and try to detect on what side users are with respect to current political questions, as long as these are two-sided questions, so that the three labels mentioned above can be meaningfully applied.

From an engineering viewpoint, it is desirable that a stance detection system be able to work not just for one target but for many different ones, or that it can be easily adapted to new ones. This means that a system should acquire knowledge of general evaluative language that speakers use to communicate a positive or negative attitude. But notice that recognizing such language is not always sufficient to solve the problem. Here is an example from Mohammad et al. (2016, p. 32), a tweet that is part of a debate about the target 'Donald Trump':

Jeb Bush is the only sane candidate in this republican lineup.

In order to calculate the stance toward the target (Trump), one needs to recognize the positive evaluation, identify its target, and then to infer that a positive stance toward Bush implies a negative stance toward Trump. This amount of reasoning power, however, exceeds that of most if not all implemented systems.

Besides microblogs, earlier CL work on stance classification has addressed domains such as internet debating forums, congressional floor debates, student essays, or public comments on proposed government regulations. The approaches tend to use generic features such as lexical sentiment (see below), and topic-specific features that have been learned from labeled data for those topics. It turned out, however, that simple classifiers with unigram or ngram features (i.e., words or word sequences taken straight from the text) are difficult to beat for these tasks (Somasundaran and Wiebe, 2010; Hasan and Ng, 2013; Mohammad et al., 2016).

5.2 Sentiment analysis

The 'sentiment analysis' or 'opinion mining' problem, which has been very popular in computational linguistics since the early 2000's, comes in different variants, one of which is quite similar to stance detection. But let us first look at so-called 'fine-grained' sentiment analysis, which, according to Liu (2002) aims at extracting from a text the following set of information units: the *holder* of the opinion, its *target* entity, an *aspect* of that entity, the *sentiment (opinion)* expressed on that aspect, and the *time* when the opinion was expressed. Product reviews are probably the most widespread domain for this application, and we can illustrate the task for a (fictitious) short text about a (fictitious) camera:

My sister bought a Ninon TT-3 yesterday. She told me that the sensor is just fantastic and beats the entire competition.

The opinion holder is the sister of the writer; the target is Ninon TT-3; the aspect is the sensor; the sentiment is (in a coarse-grained analysis) positive; and the time is probably one day prior to the writing time of the text.

Doing this fine-grained analysis automatically is difficult, and therefore several less complex variants of the task, which nonetheless may be useful for practical purposes, are being explored. Often the opinion holder and the time are not needed for a concrete application scenario (where, for instance, a manufacturer wants to get a rough overview of customer reactions). Being able to compute the aspect is desirable yet often very difficult, and therefore omitted. Finally, the target is sometimes known beforehand (for example when the texts are taken from one specific Amazon product review page) and need not be extracted. Hence, in the simplest case we are left with the task to just compute the opinion, usually called the *polarity* of the review. And if polarity is not modelled as an elaborate scale but just as the triple (negative, neutral, positive), then this version of sentiment analysis corresponds to stance detection as we have characterized it above, with the proviso that sentiment analysis is typically targeting entities (such as products), whereas stance is more generally computed for propositional targets.

Sentiment analysis systems exploit the presence of words that convey positive (*great*) or negative (*bogus*) judgement. Simplifying a little, there are basically two ways to build such a system: One can train simple *bag of words* models (recall the unigram and ngram models mentioned above) on sentences or texts whose overall polarity is known, for instance on product reviews that have an associated numerical rating. The resulting model is likely to be somewhat noisy, and it is not clear how the system will perform on texts from a different domain. The other approach involves human intervention in building a sentiment lexicon. For instance, Taboada et al. (2011) describe the SO-CAL system, which operates with a lexicon whose words have been ranked on a scale from -5 to +5 via crowdsourcing. Furthermore, SO-CAL checks whether the context of a word in a sentence calls for shifting its lexical polarity. The best-known example is the presence of negations, but the system also handles amplifiers (*particularly nice*) and downtoners (*slightly boring*), as well as so-called irrealis markers (*the movie could have been great*). Taking possible combinations of those phenomena into account, SO-CAL computes a contextual polarity for sentences and texts. The evaluation shows that the system generalizes quite well to product domains that were not part of the development data.

As indicated, this dichotomy between harvesting word lists by machine learning on the one hand and manual editing of dictionary entries on the other hand is simplifying the picture to some extent. For one thing, there are methods of semi-automatic lexicon construction that may contain translations from lexicons in other languages and/or human post-editing; for another thing, machine learning approaches can do considerably smarter things than just gathering the lexical material for sentiment lexicons. For example, one influential early system by Socher et al. (2013) learned how to construct a sentiment analysis for a sentence by passing information throughout its syntactic parse tree; and various other elaborate approaches have been proposed since then.

Aside from the mainstream sentiment analysis tasks just sketched, in the past years a smaller research community has been concerned with another very fine-grained formulation of the problem, which is called ‘entity-level sentiment’. Consider:
Federer narrowly lost the match against Nadal, who had been fervently supported by the audience.
There is little point in computing an overall polarity for a sentence like this, which does not convey a uniform polar opinion. At the same time, it clearly presents ‘good news’ or ‘bad news’ for the various parties involved, and this type of clause-level sentiment ‘flow’ among entities is being studied. Verbs play a central role here, and thus the development of task-specific verb lexicons is one of the main goals (e.g., Deng/Wiebe 2014, Klenner/Amsler 2016). For our example, *to lose against* assigns negative polarity to the subject of the clause (here: Federer) and positive polarity to the object of the preposition (here: Nadal). This assignment is independent of subjective opinions. In contrast, *to support* conveys that the subject has a positive attitude toward the direct object – and this attitude need not coincide with that of the author, who merely reports it. For the example, this means that a system should record that the audience was in favour of Nadal. A final, relatively complicated inference step for an automatic system is to derive that the audience will be pleased, because the player they supported actually won the match.

Needless to say, in even more complex sentences, modelling such sentiment flows can become quite intricate. However, the lexicons mentioned above, in tandem with a set of inference rules, provide promising first steps for this field of analysis.

6. Sentiment and stance in argumentation mining

As argumentation is revolving around the notions of 'pro' and 'contra', there is obviously a close connection between argumentation mining and stance analysis, albeit not a trivial one, as we will see. In this section, we mention some research that employed stance for solving certain subtasks of argumentation mining.[7] We aim to situate this work in a coherent larger picture (see Fig. 2), and will thereafter examine the role that sentiment analysis has played in computational argumentation mining so far, and how it could be developed in the future.

Figure 2 suggests a hierarchy of entities that play important roles in argumentation analysis and in automatic mining. For illustration, we instantiate it with some (freely made-up) notions from the realm of vaccination. We see a 'topic' as a phrase characterizing the object, person, or situation that the argumentation is about. A 'standpoint', as commonly defined, constitutes a proposition that others might agree or disagree with. Many standpoints can be formulated as opinions on certain aspects of a topic; we show here just two examples. Claims are more specific statements, and they either support or attack the standpoint; this corresponds to a positive or negative stance, indicated here by '+' and '-', respectively. Finally, the 'premises' in turn support or attack a claim, and likewise they express a stance toward the claim. - Notice that the figure is not intended as an alternative annotation scheme; it merely serves as a background for the following descriptions. (For many practical purposes, the distinction made here between 'standpoint' and 'claim' may not be relevant.)

In our first argumentation mining scenario - mapping the elementary units of a text to a structural representation - the Peldszus/Stede (2013) scheme used in the 'microtexts' corpus makes use of just claims and premises, and support/attack relations. (But it allows for recursion by means of statements that serve simultaneously as premise and claim.) The 'persuasive essay' scheme adds the next level up: our 'standpoint' corresponds to the single 'main claim' in a student essay, and various subordinate claims can be stated and defended in the essay.

In the second scenario, open-domain argument construction, the work starts from a given topic or possibly from a standpoint, and then all subordinate elements in the hierarchy are to be retrieved from text collections, as we described in Section 4. If that is successful, a system has gathered enough information to build up an argumentation for the standpoint, or to participate in a debate on the topic.

[7] Several researchers have explored the „opposite direction“, i.e. using argumentation mining in order to improve sentiment analysis. For reasons of space, we do not discuss this line of work here.

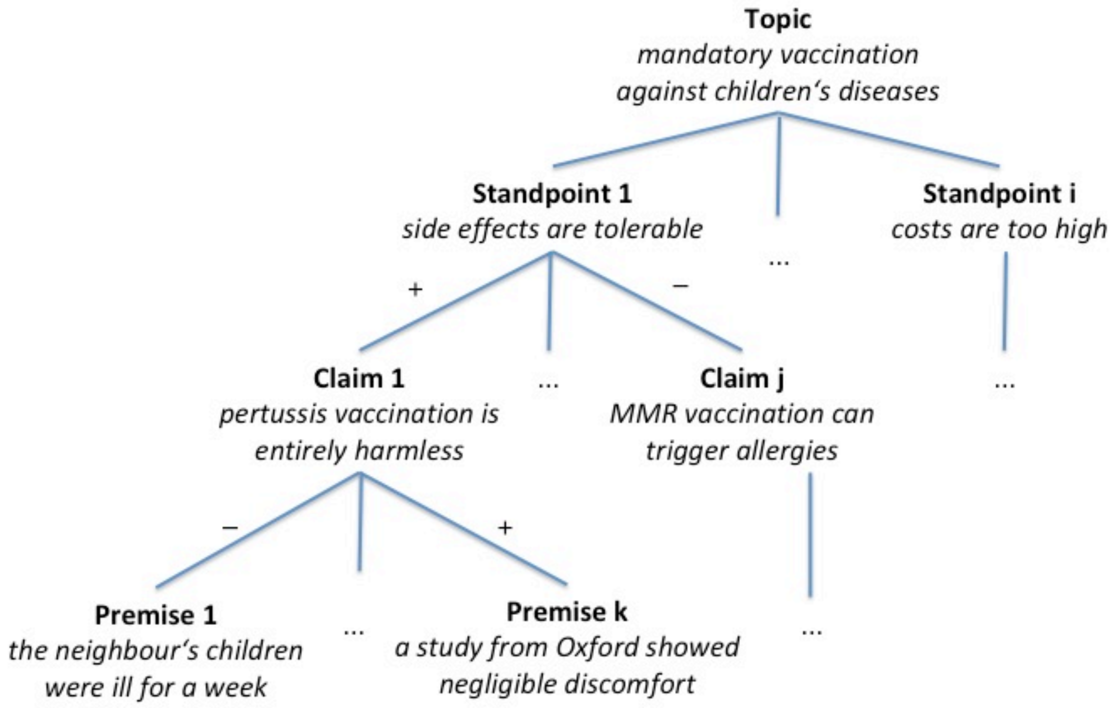


Figure 2: Entities that are subject to annotation (in text) or retrieval (from text) in argumentation mining

In this setting, where argument units are to be independently retrieved from - possibly very different - texts, stance analysis plays a very central role: A system wants to collect statements from both sides, but it is mandatory to be able to identify the two sides accurately. This can be fairly complicated, as the following example from Bar-Haim et al. (2017) illustrates.[8] Assume that for the topic 'monarchy', the system has chosen the standpoint

(1) [The monarchy]$_t$ should be abolished. ⊖

Here, a sentiment analysis should discover that the sentence expresses a negative polarity toward the target (indicated by the subscript) 'monarchy', and thus a negative stance toward the given the topic (which here is identical to the target in the sentence), indicated by the ⊖ sign. Now, the system retrieves associated claims from the text database and finds the following two:

(2) [Social traditions or hierarchies]$_t$ are essential for social order. ⊕

(3) People feel greater dignity when [choosing their head of state]$_t$. ⊕

The sentiment task is to recognize positive polarity toward social traditions and hierarchies in (2), and toward the idea of choosing a head of state in (3); these are indicated by the ⊕ signs. In the next step, analysing stance toward claim (1), the ideal

[8] The reader be warned that Bar-Haim et al. (2017) unfortunately use the term 'topic' for what we call 'standpoint' here. Our description of their work sticks to our own terminology.

system would assign ⊖ to (2) and ⊕ to 3. Hence, the sentiment polarity needs to be flipped in the case of (3) but not for (2).

We pointed out in Section 5 that automatic sentiment/stance analysis employs "visible" surface features, in conjunction with lexicons supplying prior polarities for individual words. This toolbox will not be sufficient for solving cases like the example above. For one thing, determining the sentence-level sentiment is much harder than for a sentence that, for instance, explicitly rates a product ("This meal is wonderful."). Furthermore, knowledge about the relationships between social traditions/hierarchies, choosing governments, democracy and monarchy is required to arrive at the correct stance labels - there are no surface signals to be exploited here. Bar-Haim et al. (2017) conclude that for the general task of open-domain argument construction, suitable knowledge resources are needed, as there is no realistic way to obtain the specific information via machine learning from corpora. They furthermore note that stance detection cannot generally rely on sentiment analysis. Consider the following claim, whose stance toward standpoint (1) is positive, but no explicit sentiment is present:
(4) The people, not the members of one family, should be sovereign.

A somewhat similar situation was addressed by Wojatzki and Zesch (2017) in their work on Twitter, mentioned above in Sct. 4.1. Their example
Bible: Infidels are going to hell!
maps to our Figure 2 as follows: The topic of the debate (as known by participants) is atheism; the tweet corresponds to a claim, which expresses positive stance toward Christianity, which in turn allows to infer the implicit standpoint *I am against atheism*, thus linking the claim to the topic.

Nonetheless, in the general field of argumentation mining, many researchers experimented with using straightforward sentiment features for solving one or more subtasks. Often, a simple lookup in sentiment lexicons is performed to determine a majority vote on the polarity of a sentence; some systems try to account for negations in order to avoid obvious mistakes. More elaborate analyses have to our knowledge not been employed yet. The sentence polarity is commonly added to the set of features used for computing the argumentative role of a sentence in a text. However, Afantenos et al. (2018) report that in their experiments on predicting argument structures on microtexts, sentiment features turned out to be not useful, and hence were discarded. This might be explained by the fact that microtexts do *per se* not contain sentences that do not belong to the argumentation (and thus could be surmised to be 'objective'); and for distinguishing claims and premises, the presence of a sentiment or the polarity is generally not helpful, as the 'monarchy' example has illustrated.

For longer text, where one needs to distinguish argumentative from non-argumentative sentences, sentence-level polarity might be helpful. One should be aware, however, that sentiment features are then being used as a shortcut for classifying ‘subjectivity’ in a more general sense: Not all subjective utterances are characterized by polarity; speculations or prognoses (“There will be rain tomorrow.” / “Trump is going to be re-elected.”) can clearly be claims or premises in an argument but do not convey any polarity.

One interesting application of sentiment analysis beyond the “standard” subtasks of argumentation mining is presented by Wachsmuth and Stein (2017). They are interested in the rhetorical motivations for linearizing argumentative text in particular ways, and propose to analyse a text as a "flow of 'task-related rhetorical moves'", where rhetorical effects result from different types of information underlying flows, and flows can be grouped into protoypical patterns. Any flow analysis rests on a segmentation of the text into minimal units, to which numerical or categorical labels can be ascribed. For hotel reviews, they use the sentiment polarity of units as a basis for comparing different texts in terms of moving between positive, negative and neutral units. Again, to be generalizable to other argumentative texts, sentence polarity should be replaced by stance toward the claim, which would be the more appropriate level of rhetorical analysis – but generally hard to compute.

We close this section by discussing the potential role of fine-grained entity-level sentiment analysis for argumentation mining, using an example from the 'microtexts' corpus. It requires the background knowledge that the data was collected in Berlin/Germany.

Should we continue to separate our waste for recycling?
(1) Yes, it's annoying and cumbersome to separate your rubbish properly all the time.
(2) Three different bin bags stink away in the kitchen
(3) and have to be sorted into different wheelie bins.
(4) But still Germany produces way too much rubbish
(5) and too many resources are lost when what actually should be separated and recycled is burnt.
(6) We Berliners should take the chance and become pioneers in waste separation!

First, notice that a straightforward sentence-/text-level sentiment analysis would recognize the many negative-polarity words (*annoying, cumbersome, rubbish, stink, lose*) in comparison to very few positive-polarity words and thus conclude that the text expresses negative stance toward the question – which is obviously not true.

A more elaborate analysis that includes the entity level (which is somewhat beyond the capabilities of current systems) could proceed as follows. (1) expresses negative polarity toward the target notion *separate the rubbish*. Knowing the lexical similarity between *rubbish* and *waste* leads to inferring a negative stance toward the question. (2) is explicitly negative, too, but the target (three bags) needs to be interpreted as emphasizing the quantity, which (3) reinforces. The *but* in (4) signals a contrast, and this holds not on the level of propositions but on the level of evaluation: a negative attitude in (1/2) is now turned into a positive one. This can be computed on the entity-level: *too much rubbish* has negative lexical polarity and *to produce* projects this to the subject, which is *Germany*, which the reader needs to interpret as *we*. Hence there is a negative effect for *us*. The first clause of (5) continues the pattern: *resources* is lexically positive, and the meaning of *to lose* states that a positive object projects a negative effect on the subject (you don't want to lose something precious). The second clause specifies a precondition for the first to hold, and the interpretation would require background knowledge on the connections between recycling waste und burning waste. (6) picks up the positive stance from (4) and (5) with the lexically-positive expressions *take the chance* and *become pioneers*, which address the target *waste separation*, and hence the sentence is an explicit answer to the question.

7. Conclusions

We provided a short overview of applications and subtasks of argumentation mining, mentioning two possibilities for further reading that go into more detail (see footnote 1). Throughout the survey, we made a broad distinction between two different perspectives of argumentation mining: that of extracting the components of an argument from a given text, and that of collecting these components individually from large collections of text, such that they address a desired topic.

While the technical implementations are not the focus here, we sketched the idea of building automatic classifiers that learn from text examples with annotated features, which in turn have to be defined by the system builder (but then are automatically extracted from the texts by suitable tools). In the field of argumentation mining, this has been the most common way of approaching the tasks, but as in other areas of computational linguistics, the alternative technique of neural networks is rapidly gaining ground. Here, features need not be defined; instead, the network tries to identify relevant patterns itself. While this is obviously attractive, common disadvantages are a need for more labelled traning data, and an intransparency of the processing: it is very difficult to see why a neural network made a certain classification decision – whereas in a traditional feature-based approach, their responsible configuration can be inspected.

In the second half of the paper, we connected argumentation mining to a (small) part of semantic/pragmatic analysis, viz. sentiment and stance detection. These tasks have been popular in CL for a long time now, and they are often routinely employed in the "feature pool" of systems tackling argumentation mining tasks. We pointed out that stance is indeed closely related to argument, but computing it correctly can be a difficult endeavour. On the other hand, the straightforward sentiment systems based on pre-stored lexical polarities are of only limited use, and can often actually be misleading.

At the end, we discussed the possible role of fine-grained entity-level sentiment analysis, which is a relatively new branch within the field. It amounts to a formalisation of specific semantic knowledge, in particular of the polarity-propagation behaviour of verbs (but other linguistic phenomena invite similar treatment). While large-coverage resources of this kind do not exist yet, we believe that their development and improvement could lead to significant progress for various CL applications, including argumentation mining, where robust connections to semantic analysis – of different types – is urgently needed in order to be able to handle more of the many cases of implicit, non-superficial, inference-prone phenomena that we encounter in everyday argumentative text.

Acknowledgements

Many thanks to Maria Skeppstedt and to the anonymous reviewers for their helpful comments on an earlier version of this paper.

References

Afantenos, Stergos, Andreas Peldszus, and Manfred Stede. (2018) Comparing decoding mechanisms for parsing argumentative structures. *Journal of Argumentation and Computation* 9(3), 177-192

Al-Khatib, Khalid, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, Benno Stein (2016) A News Editorial Corpus for Mining Argumentation Strategies. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp. 3433–3443, Osaka, Japan

Boltuzic, Filip and Jan Snajder (2014). Back up your stance: Recognizing arguments in online discussions. In Proc. of the First Workshop on Argumentation Mining, pp. 49–58, Baltimore, Maryland

Boltuzic, Filip and Jan Snajder (2016) Fill the gap! Analyzing implicit premises between claims from online debates. In Proc. of the Third Workshop on Argumentation Mining, pp. 124–133, Berlin, Germany

Cohen, Robin (1987) Analyzing the Structure of Argumentative Discourse. *Computational Linguistics* 13(1-2), 11-24

Deng, Lingjia and Janyce Wiebe (2014) Sentiment propagation via implicature constraints. Proc. of the 14th Conference of the European Chapter of the ACL, pp. 377-385, Gothenburg, Sweden

Eckle-Kohler, Judith, Roland Kluge, and Iryna Gurevych (2015) On the role of discourse markers for discriminating claims and premises in argumentative discourse. In Proc. Empirical Methods in Natural Language Processing, pp. 2236–2242, Lisbon, Portugal

Fellbaum, Christiand (ed.) (1998) *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Feng, Vanessa Wei and Graeme Hirst (2011) Classifying arguments by scheme. In Proc. Association for Computational Linguistics: Human Language Technologies, pp. 987–996, Portland, Oregon

Freeman, James B. (2011) *Argument Structure: Representation and Theory*. Argumentation Library, Springer

Habernal, Ivan and Iryna Gurevych (2017) Argumentation mining in user-generated web discourse. *Computational Linguistics* 43(1), 125-179

Hasan, Kazi Saidul and Vincent Ng (2013) Stance classification of ideological debates: Data, models, features, and constraints. In Proc. of the Sixth International Joint Conference on Natural Language Processing, pp. 1348–1356, Nagoya, Japan

Klenner, Manfred and Michael Amsler (2016) Sentiframes: a resource for verb-centered German sentiment inference. In: Proc. of the Tenth International Conference on Language Resources and Evaluation (LREC), Portoroz, Slovenia

Lawrence, John and Chris Reed (2017) Using Complex Argumentative Interactions to Reconstruct the Argumentative Structure of Large-Scale Debates. In 4th Workshop on Argumentation Mining, pp. 108-117, Copenhagen, Denmark

Lippi, Marco and Paolo Torroni (2016) Argumentation mining: State of the art and emerging trends. ACM Transactions on Internet Technology, 16(2):10:1–10:25

Liu, Bing (2012) *Sentiment analysis and opinion mining*. Synthesis Lectures on Human Language Technologies. San Rafael (CA): Morgan & Claypool

Mohammad, Saif M., Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, Colin Cherry (2016) SemEval-2016 Task 6: Detecting Stance in Tweets. In Proc. of SemEval 2016, pp. 31-41, San Diego, California

Peldszus, Andreas and Manfred Stede (2013) From argument diagrams to argumentation mining in texts: A survey. *Int'l Journal of Cognitive Informatics and Natural Intelligence* (IJCINI) 7(1):1–31

Peldszus, Andreas and Manfred Stede (2016) An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proc. 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*, pp. 801–816. College Publications, London

Persing, Isaac and Vincent Ng (2015) Modeling argument strength in student essays. In Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int'l Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 543–552, Beijing, China

Shnarch, Eyal, Ran Levy, Vikas Raykar, and Noam Slonim (2017) Grasp: Rich patterns for argumentation mining. In Proc. Empirical Methods in Natural Language Processing, pp. 1356–1361, Copenhagen, Denmark

Skeppstedt, Maria, Andreas Peldszus, and Manfred Stede (2018) More or less controlled elicitation of argumentative text: enlarging a microtext corpus via crowdsourcing. In Proc. of the 5th Workshop on Argumentation Mining, pp. 155–163. Brussels, Belgium

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng and Christopher Potts (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proc. of the Conference on Empirical Methods in Natural Language Processing, pp. 1631-1642, Seattle, WA

Somasundaran, Swapna and Janyce Wiebe (2010) Recognizing stances in ideological online debates. In Proc. of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 116–124, Los Angeles, CA

Stab, Christian and Iryna Gurevych (2014) Annotating argument components and relations in persuasive essays. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics, pp. 1501–1510, Dublin, Ireland

Stab, Christian and Iryna Gurevych (2017) Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–660

Stede, M., Schneider, J. (2018) *Argumentation Mining*. Synthesis Lectures on Human Language Technologies, Vol. 40. San Rafael (CA): Morgan & Claypool

Taboada, Maite, J. Brooke, M. Tofiloski, K. Voll, M. Stede (2011) Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2):267-307

Toulmin, Stephen (2008). The layout of arguments. In Jonathan E. Adler and Lance J. Rips (eds.): Reasoning: Studies of human inference and its foundations, pp. 652–677 Cambridge University Press

Wachsmuth, Henning, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein (2017a) Building an argument search engine for the web. In Proc. of the 4th Workshop on Argumentation Mining, pp. 49–59, Copenhagen, Denmark

Wachsmuth, Henning , Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein (2017b) Computational Argumentation Quality Assessment in Natural Language. In Proc. European Chapter of the Association for Computational Linguistics, pp. 176–187, Valencia, Spain

Wojatzki, Michael and Torsten Zesch (2016) Stance-based Argument Mining – Modeling Implicit Argumentation Using Stance. In Proc. of the German Conference on Natural Language Processing KONVENS, Bochum, Germany