An Object-oriented Model of Role Framing and Attitude Prediction

Manfred Klenner
Computational Linguistics, University of Zurich, Switzerland
klenner@cl.uzh.ch

1 Introduction

According to Rashkin et al. (2016) and other researchers, a text not only reveals the attitudes of the discourse referents dealt with in the text, but it also indicates how they are framed by the text author. Do they play a positive or negative role in the exposed narrative? We call this the author's role framing. While the attitudes tell us who is for or against who (pro and con relations), the role framing indicates who is a beneficiary or a victim etc. of the situation described. Both, attitude and role framing depend on the factuality status of the described situations. For instance, if A hopes that B helps C, than A has a positive attitude towards C, but neither is C is beneficiary nor is there an attitude of B towards C. The reason is that *hope* is a non-factive verb and *help*, thus, non-factual. If we were told that A *might* hope that B helps C, nothing follows at all, since now even *hope* in non-factual.

Not only verbs like *hope*, but also nouns like *destruction* and predicatively used adjectives like e.g. *angry* in *Trump is angry that our neighbors are hurting Wisconsin and other border states* indicate attitude. Here, Trump and Canada (neighbors) are adversaries, but Trump is pro Wisconsin, which is framed as a victim. Moreover, constructions like *It is sad that ..* clearly reveal the stance of the author towards an event and - as a consequence - towards the participants of this event. So in the end, we are able to derive the author's stance, the proclaimed attitudes among referents and the role framing that comes with a text. We introduce an object-oriented model realized in Python to deal with these topics. An empirical evaluation is presented concerning predicatively used adjectives.

2 Attitude Prediction and Role Framing

A verb might express an attitude between an opinion source and target role. It also sometimes frames a particular role in terms of a positive or negative effect the filler of the role receives in an affirmative usage of the verb. If A helps B, A has a positive attitude towards B and B is a beneficiary since he actually receives help. The author is committed to the truth of what he writes and the reader is entitled to draw conclusions on that basis. Verbs, nouns and adjectives subcategorizing for a complement clause pose a commitment on it regarding the truth of the subclause. Among others, Nairn et al. (2006) and Karttunen (2012) have described the different kinds of verb-specific commitments. They called it *verb implicatures*. The crucial point is that truth commitment and negation give rise to factuality, counterfactuality and non-factuality. While non-factuality blocks any inference, counterfactuality inverts the attitude and the effects of the verb. Our model does not explicitly ascribe factuality, we just use the equivalent notion of truth commitment in combination with negation. In sum, the attitude expressed and the role framing depends on the meaning of the matrix verb, its truth commitment on the subclause and the affirmative status of both, the matrix verb and the subclause verb. This is, mutatis mutantis, true for nouns and predicatively used adjectives as well.

There is wide range of phenomena, from verbs that are falsehood committing if affirmatively used and truth committing if negated (*forget*) to verbs with opposite commitments (*manage*). There are factive verbs which commit the truth in any case and non-factive verbs that do not cast a commitment at all (truth or falsehood). See e.g. Karttunen (2012) for the details. Keep in mind, that this not only holds for verbs, but for nouns and adjectives as well. For the ease of presentation, however, we explicate the underlying principles with reference to verbs.

The intuition behind our approach for attitude prediction and role frameing can be illustrated by a couple of simple examples: if A is *against* something that is bad/good for B, then A is for/against B. If A is *for* something that is bad/good for B, then A is against/for B. This is true for the affirmative usage. In order to deal with negation, we just switch the labels (invert them): A *against* (matrix) verb is turned into a *for* verb if negation is present and vice versa. A *bad-for* (subclause) verb is turned into *good-for* if negation is present and vice versa.

We call predicates subcategorizing for a clausal complement *disapprove* predicates if they express an opposition of the opinion holder towards the subclause event (e.g *criticize, is_shocked*). The complementary class are *approve* predicates (e.g. *hope,is_amused*). Verbs with a positive or negative effect on the patient role (e.g. *promote,hurt*) are called *good-for* and *bad-for* verbs, cf. Deng and Wiebe (2015).

Here is an example with negation: A appreciates that B was not promoted. *Appreciate* is an approve verb, *promote* is a good-for (patient) verb, but since negated this turns into a bad-for verb. Approving something that is bad-for someone means to have a negative attitude towards them.

Table 1 shows the composition principles applied to subclause embeddings where the matrix verb is a disapprove verb (e.g. *regret*) and the subclause verb is a good-for verb (e.g. *promote*). An affirmative use (aff == 1) of both, the matrix and the subclause verb leads to a con relation (A regrets that B gets promoted). Negation (aff == 0) switches the value (see the *derived* columns in Tab.1: disapprove $\rightarrow$ approve and good-for $\rightarrow$ bad-for). Any other constellation (e.g. an approve matrix verb and a bad-for subclause verb) can easily be derived from that table.

#	matrix	aff	derived	subclause	aff	derived	attitude
1	A disapproves	1	A disapprove	good-for B	1	good-for B	A con B
2	A disapproves	1	A disapprove	good-for B	0	bad-for B	A pro B
3	A disapproves	0	A approve	good-for B	1	good-for B	A pro B
4	A disapproves	0	A approve	good-for B	0	bad-for B	A con B

Table 1: Principles of the Combination of a Disapprove and a Good-for Verb

We use the verb lexicon of Klenner et al. (2017). For each verb frame we thus know whether it is an approve or disapprove verb (if it takes a complement clause) or whether it is a good-for or bad-for verb (verbs with a patient role). For clauses subcategorizing verbs moreover their commitment (truth, falsehood, no commitment) is specified - for both, the affirmative and the negated usage. We have modelled about 60 German adjectives in order to evaluate our object-oriented prototype. Our system consists of a dependency parser (cf. Sennrich et al. (2009)) and rules for predicate argument extraction[1].

3 An Object-oriented Model in Python

A sentence is represented in Python by dictionaries where the grammatical functions are the keys. Fig. 1 shows the representation for the example sentence *The senate is shocked that the minister helps the president to cheat the people*[2]. For readability, subclauses get their own dictionary variable (c2 and

```
s={'subj':'senate','main':'is_shocked','neg':'no','comp':c2}
c2={'subj':'minister', 'verb':'help','neg':'no','obj':'president','comp':c3}
c3={'subj':'president', 'verb':'cheat','neg':'no','obj':'people'}
```

Figure 1: Representation of the Input Sentence as a Python Dictionary

[1]In this paper, we use grammatical functions instead of semantic roles.

[2]We provide an English example, the German version works much like the English one.

c3). The interpretation is triggered by instantiating the main predicate of the matrix clause. We call: eval(s['main'])(s,'t') where *s* is the sentence and *t* means truth committed (affirmative use, no modals present). Interpretation is an outside-in instantiation of the predicates denoted by the verbs, nouns and adjectives of the clauses. Each predicate class has methods to determine attitude and role framing. The polar roles *opinion source* and *opinion target* are set depending on the predicate: two variables, source and target, are used to store the particular grammatical functions that realize these polar roles. Fig. 2

```
class is_shocked(disapprove):
1    def __init__(self,sent,com):
2        self.sc=eval(sent[’comp’][’verb’])(sent[’comp’],’t’)
3        self.attitude(sent,self.sc,com)
```

Figure 2: Definition of *is_shocked*

shows the definition of the disapprove predicate *is_shocked*. Both, the disapprove and approve class set the (logical) subject to be the opinion source (i.e. source = 'subj'). The init method has (besides self) two parameters, the sentence and a truth commitment. The commitment that *is_shocked* casts on its subclause is - in any case - *t* (truth commitment), since *is_shocked* is factive. We, thus, directly instantiate (line 2) the subclause verb (sent['comp']['verb'] gives *help*) with the whole subclause (i.e. sent[comp]) and *t* as truth commitment. This call leads to the evaluation of the embedded clause (*help*). The attitude expressed by *is_shocked* is predicted by the method *attitude* in line 3 (self.sc is the instance of *help*, providing access to the source and target of that verb instance).

```
class help(approve, atnan, goodfor):
1    def __init__(self,sent,com):
2        self.sc=eval(sent[’comp’][’verb’])
3                        (sent[’comp’],self.propagates(sent[’comp’][’neg’],com))
4        self.attitude(sent,self.sc,com)

class atnan:
    def propagates(self,neg,com):
1        if com == ’n’: return ’n’
2        elif com == ’f’ and neg == ’yes’: return ’t’
3        elif com == ’f’ and neg == ’no’:  return ’n’
4        elif com == ’t’ and neg == ’no’:  return ’t’
5        elif com == ’t’ and neg == ’yes’: return ’n’
```

Figure 3: Definition of *help*(upper part) and of *propagate*(lower part)

Fig. 3 shows the definition of the approve verb *help*. The opinion target is given by the direct object. This is the contribution of the class *goodfor* (i.e. target='obj', not shown). The truth commitment *help* casts on its subclause depends on the truth commitment it receives from the embedding predicate (here *is_shocked*) and the verb signature and the affirmative status of *help*. The method *propagate* (see Fig. 3 lower part) determines this. *com* is the commitment of the matrix predicate, while *neg* refers to the affirmative status of the embedded verb, here *help*. The class *atnan* models how the truth commitment of the embedding predicate influences the truth commitment of the embedded predicate given that it is truth committing if affirmative (at) and has no commitment if negated (nan, thus the class *atnan*).

If *help* is embedded under *n* (*com* == 'n') which means *no commitment*, like in *A hopes that B helps to free C*, *n* is returned (line 1), i.e. *free* receives no commitment. If it is embedded under negation with a falsehood commitment *f* (line 2) like in *A refuses not to help to free c*, *t* is returned, *free* is truth committed and so on. How commitments propagate depend on the verb (class). According to line 5, no commitment is set if the predicate (*help*) is negated and embedded in a truth committing verb like in *A criticizes that B not helps C to free D*. (i.e. it is unknown whether D is freed by C or not). This is in contrast to verbs like *manage* where the same constellation leads to *f* (*A criticizes that B not manages to free D*).

```
class cheat(badfor):
    def __init__(self,sent,com):
        if com=='t' and sent['neg']=='no': negactor(sent[self.source])
        self.direct-attitude(sent,com)
```

Figure 4: Definition of *cheat*

Fig. 4 shows the definition of *cheat*, a *badfor* verb. The truth commitment wrt. our example is 't' and since it is used affirmatively (i.e. it is factual) the source (self.source = president) is classified as a negative actor (*negactor*). *cheat* expresses a direct (negative) attitude between the source and the target.

```
class badfor:
    def direct-attitude(self,sent,com):
        if sent['neg'] == 'no' and com == 't':
            con(sent[self.source],sent[self.target])
            negaffected(sent[self.target])
        elif sent['neg'] == 'yes' and com == 't':
            pro(sent[self.source],sent[self.target])
            posaffected(sent[self.target])
....
```

Figure 5: Definition (partial) of *badfor*

The method *direct-attitude* of *badfor* (see Fig. 5) sets a con relation between the source and target in an affirmative context (neg=='no') under a truth commitment 't'. The target then receives a negative effect (*negaffected*). If negated (still under 't'), a pro relation is set and the target is said to be positively affected (*posaffected*). *pro, con, negaffected and posaffected* are for bookkeeping, they store instances and relations.

```
class disapprove(subj-source):
    def indirect-attitude(self,sent,sc,com):
1        if com=='t' and sent['neg'] == 'no'
2            con(sent[self.source],sent[self.comp][sc.source])
3            if sent[self.comp]['neg'] == 'yes':
4                 if isinstance(sc,goodfor):
5                     pro(sent[self.source],sent[self.comp][sc.target])
6                 if isinstance(sc,badfor):
7                     con(sent[self.source],sent[self.comp][sc.target])
8            elif sent[self.comp]['neg'] == 'no':
9                 if isinstance(sc,badfor):
10                    pro(sent[self.source],sent[self.comp][sc.target])
11                if isinstance(sc,goodfor):
12                    con(sent[self.source],sent[self.comp][sc.target])
....
```

Figure 6: Definition (partial) of *disapprove*

While goodfor and badfor verbs express a direct attitude of the verb's source towards the verb's target, approve and disapprove predicates cast an indirect attitude of the source of the matrix predicate on the source (and the target) of the subclause predicate. Fig. 6 shows the (partial) definition of *disapprove*, which inherits from the class *subj-source* (not shown), which just tells that subject is the opinion source (source='subj'). The disapprove class defines the class method *indirect-attitude*. It serves two purposes: the attitude between the source of the matrix predicate and the source of the subclause is determined (here e.g. *con*, line 2). In our example, the disapprove predicate *is_shocked* yields a con relation (line 2) with sent[self.source]=senate and sent[self.comp][sc.source]=sent[self.comp][subj]=minister which establishes: con(senate,minister). If the embedded verb is an instance of goodfor (i.e. has a target, not in our case), then a pro relation between the source (of the matrix predicate) and that target (of the subclause predicate) is set like in e.g. *A criticizes that B not promotes C*. In sum: we get a negactor (president), a negaffected entity (people), three con relations (senate, minster), (minister,people) and (president,people) and a pro relation (minister,president). Other attitudes, namely con(senate,president) and pro(senate,people) are indirectly derivable, e.g. by rules like con(?x,?y) and con(?y,?z) then pro(?x,?z) which gives us pro(senate,people).

4 Related Work

The goal of the rule-based approach of Deng and Wiebe (2014), Deng et al. (2014) and Deng and Wiebe (2015) is to detect entities that are in a positive (PosPair) or negative (NegPair) relation to each other. Rules are realized in the framework of Probabilistic Soft Logic, where the rule weights depend on the output of the preprocessing pipeline, i.e. two SVM classifiers and three existing sentiment analysis systems. The model of Deng and Wiebe (2015) also copes with event-level sentiment inference, however truth commitment is not taken into account. Also, role framing does not play any role in their framework.

Rashkin et al. (2016) have presented an elaborate model that is meant to explicate the relations between all involved entities: the reader, the writer, and the entities referred to by a sentence. Also, the internal states of the referents and their values are part of the model. Their resource, called connotation frames (see also Choi et al. (2016)), was created by crowdsourcing , the model parameters (e.g. values for positive and negative scores) are average values. In contrast, the resources we use and the one we have additionally created are manually specified. Truth commitment etc. is not taken into account in their model.

Klenner and Clematide (2016) introduce a rule-based system for German realized with Description Logic and SWRL. The (numerous) rules also take the affirmative and factuality status of the sentence into account. The goal is to instantiate relations (con and pro) expressing the attitudes of entities towards each other. Recently, Klenner et al. (2017) have revised and augmented their model which now also induces a reader and writer perspective. Our model is meant to solve the attitude prediction and role framing tasks set by Klenner et al. (2017). It is aimed to especially solve cope with a new class of inference indicators, namely predicatively used adjectives.

5 Empirical Evaluation

The focus of this paper lies on predicatively used adjectives. We used the polarity lexicon described in Clematide and Klenner (2010) in order to generate expamples where an adjective is combined with a personal pronoun and a subclause comprising a verb from Klenner and Amsler (2016). For instance, the German version of the sentence *it is provocative that they ignore him* is part of the set of 100 sentences we used. We annotated these sentences wrt. to the attitudes between the virtual writer of such a sentence and the virtual referents (*they,him*). We did neither annotate the attitude between the referents nor the role framing (e.g. *they against him* and *victim(him)*), since these inferences come from the verbs not the adjectives. In other words, we were wondering whether the reasoning scheme that worked for verbs and nouns also worked (without modifications) for adjectives. It turned out that such a model overgenerates. Recall is 100%, but precision is about 83%. The reason is the model predicts an attitude towards the target in each and every case. Given *it is sad that they hurt him*, a positive attitude of the virtual writer of that sentence towards *him* is (correctly) predicted. But this is not so clear in the following sentences: *it is stupid that they cheat him* and *it is jaunty that they laugh at him*. Here, the focus seems to lie much more on the attitude towards the source of the subclause (here *they*). There are also interesting examples where the adjective seems to suppress the attitude normally expressed by the embedded verb. For instance, in *He is responsible that she abandoned her plans* a con relation between *she* and *plan* is incorrect. However, if we replace *responsible* with *happy* such an inference was valid. This inference blocking is not yet covered by our model.

6 Conclusions

We have introduced a prototype of an object-oriented model implemented in Python for attitude prediction and role framing. Compared to previous approaches, our model is lean and it deals not only with predictions derived from verbs and nouns, but also with predicative constructions where an adjective has an effect and a truth commitment on its subcategorized clause. Our claim is that lexical resources are still valuable and that they even are, especially for the kind of reasoning we are interested in, indispensable. Our empirical evaluation revealed that our model is in part overgeneral. We believe that crowdsourcing experiments might help to get a clearer picture of the underlying principles.

References

Choi, E., H. Rashkin, L. Zettlemoyer, and Y. Choi (2016). Document-level sentiment inference with social, faction, and discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, pp. 333–343.

Clematide, S. and M. Klenner (2010). Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, Lisbon, Portugal, pp. 7–13.

Deng, L. and J. Wiebe (2014). Sentiment propagation via implicature constraints. *Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2014).*

Deng, L. and J. Wiebe (2015). Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, pp. 179–189.

Deng, L., J. Wiebe, and Y. Choi (2014). Joint inference and disambiguation of implicit sentiments via implicature constraints. *Proceedings of COLING.*

Karttunen, L. (2012). Simple and phrasal implicatives. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, Stroudsburg, PA, USA, pp. 124–131. Association for Computational Linguistics.

Klenner, M. and M. Amsler (2016). Sentiframes: A resource for verb-centered German sentiment inference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, Portoro, Slovenia, pp. 2888–2891.

Klenner, M. and S. Clematide (2016). How factuality determines sentiment inferences. In I. T. Claire Gardent, Raffaella Bernardi (Ed.), *Proceedings of *SEM 2016: The Fith Joint Conference on Lexical and Computational Semantics*, Berlin, Germany, pp. 75–84.

Klenner, M., D. Tuggener, and S. Clematide (2017). Stance detection in Facebook posts of a German right-wing party. In *Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem).*

Nairn, R., C. Condoravdi, and L. Karttunen (2006). Computing relative polarity for textual inference. In *Proceedings of Inference in Computational Semantics (ICoS 5)*, Buxton, England, pp. 67–75.

Rashkin, H., S. Singh, and Y. Choi (2016). Connotation frames: A data-driven investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, pp. 311–321.

Sennrich, R., G. Schneider, M. Volk, and M. Warin (2009). A new hybrid dependency parser for German. In *Proceedings of the German Society for Computational Linguistics and Language Technology (GSCL)*, Potsdam, Germany, pp. 115–124.