Supervised opinion frames detection with RAID

Alessio Palmero Aprosio[1], Francesco Corcoglioniti[1], Mauro Dragoni[1], and Marco Rospocher[1]

Fondazione Bruno Kessler, Trento, Italy
[aprosio,dragoni,corcoglio,rospocher]@fbk.eu

Abstract. Most systems for opinion analysis focus on the classification of opinion polarities and rarely consider the task of identifying the different elements and relations forming an opinion frame. In this paper, we present RAID, a tool featuring a processing pipeline for the extraction of opinion frames from text with their opinion expressions, holders, targets and polarities. RAID leverages a lexical, syntactic and semantic analysis of text, using several NLP tools such as dependency parsing, semantic role labelling, named entity recognition and word sense disambiguation. In addition, linguistic resources such as SenticNet and the MPQA Subjectivity Lexicon are used both to locate opinions in the text and to classify their polarities according to a fuzzy model that combines the sentiment values of different opinion words. RAID was evaluated on three different datasets and is released as open source software under the GPLv3 license.

1 Introduction

In the last years, analysis of sentiment and emotions in texts got increasing attention in the research community, and big companies started to release commercial tools whose purpose is to analyze opinions in products reviews, blog posts and social contents. See, for example, business tools such as IBM Watson Analytics[1] and SenticNet,[2] or academic tools like Stanford Sentiment tool[3].

Unfortunately, most of the commercial tools available for sentiment classification are limited to a small set of emotions, and can only manage explicit expressions, without being able to understand implicit opinions. In particular, they cannot deeply understand the frame outside the opinion expression itself, including the identification of the different roles involved in the expression. In contrast with this, opinion mining techniques capture, along with the sentiment expression, the subject(s) and the object(s) of the opinion, and its strength (intensity). This paradigm, which is the focus of this work, has great potential in gathering political trends, brand perception and business intelligence.

Given a sentence, our task is to identify each *opinion frame* in it, extracting its *expression* span, *polarity*, *holder* and *target* text spans.[4] For example, in the sentence:

[1] http://www.ibm.com/analytics/

[2] http://business.sentic.net/

[3] http://nlp.stanford.edu/sentiment/

[4] In literature, terms defining roles in opinions may vary: in particular, the holder can also be expressed as *source*, and the target as *topic*.

> *Conservative Justice Minister Kenneth Clarke said Britain's exit from the EU would be disastrous.*

token 'disastrous' is an expression that clearly denotes an opinion whose polarity is negative, holder is 'Minister Kenneth Clarke' and target is 'Britain's exit from the EU'.

Early works on opinion classification used very simple lexical features [18], following the general idea that adding complex (and computationally expensive) features leads to a small increment of performances. This approach worked well for sentiment classification, but is not enough powerful when the task consists in extracting the whole opinion frame with its expression, holder, target and polarity. For this complex task, which involves relations between entities, tools for *deep* Natural Language Processing (NLP) such as dependency parsing and semantic role labelling (SRL) are used due to their capability of extracting semantic relations between entities mentioned in texts.

In this paper, we present RAID, a tool for identifying opinion frames in texts leveraging deep NLP and semantic features extracted from text. RAID extraction algorithm consists of a number of processing steps organized in a pipeline:

- first, we use a Conditional Random Field (CRF) tagger to identify the opinion expressions in a sentence, using features extracted from NLP tools and resources such as SenticNet [3] and the MPQA Subjectivity Lexicon;
- then, target(s) and holder(s) for each expression are extracted using a combination of Support Vector Machine (SVM) classifiers, employing features that convey lexical, syntactic and semantic properties of the candidate target/holder and that leverage a syntactic and semantic role labelling (SRL) analysis of the text;
- finally, we classify the polarity of the expression using fuzzy logic for modeling concept polarities, combining it with a knowledge graph built on top of SenticNet.

We evaluated RAID on three different datasets using the intersection-based precision-recall measures [10], as well as the evaluation measures used in the ESWC2015-CLSA[5] challenge where RAID was a participant system. RAID is released under the GPLv3 license and is available as a module of Pikes,[6] a free knowledge extraction suite that includes also a NLP pipeline based on Stanford CoreNLP[7] and Mate Tools,[8], as well as a rule-based application capturing and formalizing in RDF important linguistic aspects, and a set of tools that allows a user to access and query common Semantic Web and NLP resources. The source code[9] and a working demo[10] of RAID are available online.

The remainder of the paper is organized as follows. Section 2 contains an overview of related work and describes the resources used for training and evaluation. Section 3 illustrates the approach for opinion expression, holder and target extraction, along with polarity classification. Section 4 reports the performances of our system over the three considered datasets. Finally, Section 5 sets out conclusions and possible future works.

[5] https://github.com/diegoref/ESWC-CLSA

[6] http://pikes.fbk.eu/

[7] http://nlp.stanford.edu/software/corenlp.shtml

[8] https://code.google.com/p/mate-tools/

[9] https://github.com/dkmfbk/pikes

[10] https://knowledgestore2.fbk.eu/pikes-demo/

2 Related work

In this section we provide a brief overview of related work in opinion frame extraction (Section 2.1) and we describe three relevant datasets annotated with opinion frames that we use for training and testing RAID (Section 2.2).

2.1 Approaches for opinion frames extraction

There are several works dealing with the extraction of opinion frames including opinion expression, holder, target and polarity.

In [5], opinion expressions are extracted using CRF-based sequence taggers and extracting the n-best sequences, while the opinion holder is identified using a Maximum Entropy relation classifier. Evaluation is performed against the MPQA corpus [15] (400 manually annotated documents at that time, see Section 2.2).

The work by Ruppenhofer et al. [12] describes how the perfect annotated resource should deal with subjective expressions, both direct and hidden (i.e. a journalist showing his idea on a particular topic). They also show how SRL can help the task, and provide some examples where SRL is not enough.

The works described in [11] and [2] deal with the problem of extracting opinions from news. In [11], the authors use a FrameNet-based semantic role labeller: if the detected frame belongs to a selected list of frames, then manually crafted mapping rules are used to map some roles to the opinion holder/topic. Instead, [2] concentrates the effort on quotations extracted from news, identifying holder, target and expression using various external resources (such as WordNet-Affect and SentiWordNet), without the help of semantic role labelling.

In [16], holder extraction is performed by using convolution kernels, by identifying meaningful fragments of sequences or trees by themselves.

Sentilo [8] extracts opinion holders, topics (the targets) and sub-topics in a sentence, where a sub-topic is an entity related to the actual main target of the opinion.

Johansson et al. [10] extract opinion expressions using relational features between different opinions contained in the text. They also increase accuracy using a reranker and evaluate the performances of their system over the MPQA corpus.

Recently, the work in [1] describes an approach that projects opinion extraction on different languages using a running system in a source language and a word-aligned parallel corpus.

A good general overview of opinion mining can be found in [21] and [4].

2.2 Datasets annotated with opinion frames

We briefly describe three datasets containing text documents manually annotated with opinion frames, summarizing in Table 1 their contents.

MPQA Opinion Corpus One of the first datasets annotated with opinion frames is the Multi-Perspective Question Answering (MPQA) Opinion Corpus [15]. In its latest version, it consists of 691 news articles from various English news sources. The corpus is manually annotated with what the authors call "private states", i.e. opinions, emotions, sentiments, speculations, evaluations, and internal states that cannot be directly observed by others. The annotations are at expression (subsentence) level and each expression is connected with its corresponding source (holder). Each source is in turn connected to a coreference chain, that can end to a real span in the text, or to the writer. Otherwise, the holder is considered as "implicit". Expressions are finally enhanced with other properties such as intensity (low, medium, high, extreme), polarity (positive, negative, neutral) and even confidence of the human annotator. Targets of opinions are annotated only in particular cases (for attitudes), but an effort in that direction can be found in [13]. The MPQA annotation scheme distinguishes between direct subjective expressions (DSE), expressive subjective expressions (ESE) and objective speech events (OSE). For example, in the sentence:

> *"The report is full of absurdities," Xirao-Nima said.*

the token 'said' is a direct subjective expression where the intensity is 'neutral', the source is 'Xirao-Nima', the attitude (polarity) is 'negative', and the topic is 'report'; the expression 'full of absurdities' is an expressive subjective expression, where the intensity is 'high', the source is 'Xirao-Nima', and the attitude is 'negative'.

News Texts with Opinion Annotations (NTOA) This dataset has been produced as part of the OpeNER EU project[11] and consists of 471 pieces of text extracted from political news. It is freely available online.[12] The annotation does not cover the whole text, but only some sentences. For each article, a sentence containing an opinion is chosen and annotated; an additional "non opinionated" sentence is selected, just to include negative examples in the set useful for training. Sentences are then annotated with opinion holder holder, target, expression (distinguishing between *direct expression of attitude* and *indirect expression of attitude*), polarity (positive, negative, neutral) and strength (normal and strong). For example, in the sentence:

> *Germany wants a looser arrangement among national bank-resolution authorities.*

the token 'wants' represents the expression, 'Germany' is the holder, 'a looser arrangement' is the target; the polarity is 'positive' and the strength is 'normal'.

Darmstadt Service Review Corpus (DSRC) The Darmstadt Service Review Corpus [14] consists of consumer reviews annotated with opinion related information at the sentence and expression levels. In particular, word spans for opinion expressions, opinion targets and holders are marked. The data consists of 474 reviews collected from various review portals, and related to the universities and online services domains. For instance, in the sentence:

> *I don't know why this site seems to attract people who have sour grapes with respect to Capella.*

[11] http://www.opener-project.eu/

[12] https://github.com/opener-project/opinion_annotations_news

the span ‘sour grapes’ is annotated as the expression, ‘people’ as holder, ‘Capella’ as target; the polarity is ‘negative’ and the strength is ‘weak’.

Table 1. Statistics about the available datasets.

Dataset	Docs	Sents	Tokens	Opinions
MPQA Opinion Corpus (DSE+ESE)	691	15,883	387,390	24,475
News Texts with Opinion Annotations	434	580	12,020	816
Darmstadt Service Review Corpus	491	9,836	177,020	2,867
Darmstadt Service Review Corpus (challenge)	372	6,221	113,293	2,014

3 The RAID pipeline

Opinion extraction in RAID consists of a number of processing steps organized in a pipeline. An input text document is pre-processed by running a number of NLP tools on it, in order to obtain the necessary NLP annotations (Section 3.1). Opinion expression spans are identified on a per-sentence basis (Section 3.2). For each identified expression, the corresponding holder and target spans are then extracted (Section 3.3). Finally, a positive / negative / neutral opinion polarity is assigned to the expression (Section 3.4). These steps are detailed in the remainder of the section.

3.1 Pre-processing

Starting from the raw document text, we apply a set of linguistic tools whose output is used to extract the features needed for opinion extraction. In particular, we use Tintop, the NLP pipeline included in the Pikes suite (see Section 1). It performs tokenization, sentence splitting, part-of-speech tagging, dependency parsing, semantic role labelling (SRL), named entity recognition, word sense disambiguation and supersense tagging with respect to WordNet 3.0. (using the UKB[13] tagger).

3.2 Extraction of opinion expressions

The task of extracting opinion expressions from a sentence is formulated as a sequence labelling problem and consists in tagging each token of the sentence as being either *inside*, *outside* or the *beginning* of an opinion expression, according to the popular IOB2 format.[14]. We use as supervised classifier a Conditional Random Field (CRF) trained with the Passive-Aggressive [6] algorithm, using the implementation provided by CRFsuite, a very fast classification tool publicly available on the author website.[15]. The features used to train the CRF include the word form, lemma and part-of-speech tag of

[13] http://ixa2.si.ehu.es/ukb/

[14] https://en.wikipedia.org/wiki/Inside_Outside_Beginning

[15] http://www.chokkan.org/software/crfsuite/

each token, as well as whether the token is included in the SenticNet [3] and Subjectivity Lexicon [17] resources. A sliding windows of size 2 is used, meaning that each token is classified using features of the two preceding and following tokens in the sentence. Finally, the gold column is added using the IOB2 format.

3.3 Extraction of opinion holders/targets

For each identified opinion expression span, the extraction of the associated holder and target spans in the enclosing sentence is done in three phases described next.

Identification of candidate holder/target head tokens The opinion expression is shrinked or enlarged, if needed, until a unique noun, verb, adjective or adverb head token e can be identified inside it.[16] The head token e is used as an anchor for identifying two sets of *candidate* tokens H_e and T_e, whose elements can possibly be the heads of holder and target spans for e, respectively. We build H_e and T_e as the largest sets satisfying the following conditions:

- tokens in H_e must be nouns or pronouns (as holders are agents);
- tokens in T_e must be nouns, pronouns or verbs (as targets can also be events);
- tokens in H_e or T_e cannot be or syntactically depend on modifier tokens, unless the token beign modified is e (e.g., in 'he likes beers from Germany' with $e =$'likes', T_e contains 'beers' but not 'Germany');
- tokens in H_e or T_e cannot be part of noun or verb phrases coordinated with e or an ancestor of e in the dependency tree (e.g., in 'he likes beer and she loves wine' with $e =$'likes', H_e and T_e cannot have tokens in 'she loves wine');
- $e \notin H_e$ and $e \notin T_e$.

Selection of holder/target head tokens Given e, H_e and T_e, we select the sets of holder head tokens $\hat{H}_e \subseteq H_e$ and target head tokens $\hat{T}_e \subseteq T_e$ as described next for target tokens (the same approach applies to holder tokens). For each token $t \in T_e$, we apply a supervised linear classifier to compute a score $s(t, e)$ that quantifies the likelihood of $t \in \hat{T}_e$, with $s(t, e) > 0$ if t is predicted to belong to $\hat{T}_e$. We train the classifier with the LIBLINEAR[17] software, using logistic regression as loss function and the score returned by the classifier as $s(t, e)$. The following features are used:

- *Lexical features*: lemmas of t and e.
- *Syntactic features*: (i) part-of-speech tag of t and e; (ii) dependency relation to parent token for t and e; (iii) verb voice of e, either active, passive, or none, if not a verb; (iv) whether t is a proper noun. (v) encoding of path p linking t to e in the dependency tree, simplified by removing COORD and CONJ coordination links; (vi) whether the length of p is not greater than c, for each $c \in 1 \ldots$ max path length.
- *Semantic features*: (i) WordNet 3.0 synsets of t and e, including hypernyms; (ii) WordNet 3.0 supersenses of t and e; (iii) BBN entity type[18] of t, if any, such

[16] This normalization does not affect the expression returned by the system and is required as expressions extracted in Section 3.2 might not be aligned with parse tree constituents.

[17] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[18] https://catalog.ldc.upenn.edu/docs/LDC2005T33/BBN-Types-Subtypes.html

as person, organization or location; (iv) path linking t to e in a semantic graph having a node for each token and an edge for each predicate-argument SRL relation (head tokens are connected), labelled with the thematic role.

To compute $\hat{T}_e$ we impose that candidate tokens in T_e that are coordinated one to another (e.g., 'beer' and 'wine' in 'they like beer and wine') are either all selected or not selected. To this end, for each cluster $C \subseteq T_e$ of coordinated tokens, and for each token $t \in C$, we alter the token score by setting $s(t, e) = \min_{t' \in C} s(t', e)$. We then compute $\hat{T}_e$ as the set of tokens having the largest non-zero score (if any), i.e., $\hat{T}_e = \{t \in T_e \mid s(t, e) > 0 \wedge \forall t' \in T_e, s(t, e) \geq s(t', e)\}$.

Selection of holder/target spans Although the selection of head tokens uniquely identifies holder and target entities in the text (e.g., 'crowd' and 'Obama' in 'the crowd acclaimed president Barack Obama'), it may be unsuitable to applications and evaluation metrics that expect the selection of longer spans of text (e.g., 'the crowd' and 'president Barack Obama'). We thus expand each selected head token to a longer span of text, using a supervised technique able to adapt to different, corpus-specific selection criteria (e.g., select 'Barack Obama' vs 'president Barack Obama').[19]

Our expansion algorithm is based on a linear SVM classifier (we use the LIBSVM[20] software package) that, given a currently *selected span* S and a disjoint *candidate span* S_c dominated by some token in S, decides whether S_c can be added to S. We apply the classifier iteratively starting from an initial selected span consisting of the holder/target head token only. At each iteration, we consider (if it exists) a topmost token t in the dependency tree that is dominated by tokens in the currently selected span S, skipping prepositions and conjunctions. We build the candidate span $S_c(t)$ for token t by including all the tokens of named entities overlapping with t (e.g., 'Barack Obama' for $t =$ 'Obama') and of auxiliary and main verb tokens belonging to the verb 'catena' of t (e.g., 'would have been' for $t =$ 'would'). We apply the classifier to S and $S_c(t)$ and, if the outcome is positive, we add $S_c(t)$ to S. The process is iterated until a fix point is reached. The employed classifier features are listed below, where S is the selected span, $S_c(t)$ is the candidate span with head token t, $S^*_c(t)$ is the span with descendant tokens of t in the dependency tree, and p is the nearest ancestor of t that belongs to S:

- *Lexical features*: lemmas of p and t.
- *Syntactic features*: (i) part-of-speech tags of p and t; (ii) whether p and t are proper nouns; (iii) dependency relations from and to t; (iv) token distance between $S_c(t)$ and S (adjacent, very near, near or far, based on number of tokens separating the spans); (v) token distance between $S^*_c(t)$ and S (same categories).
- *Semantic features*: thematic role of the SRL relation between p and t, if p is a predicate and t is the head token of an argument of p.

[19] The choice of a supervised approach in place of hard-coded rules is motivated also by observing that none of the datasets considered in Section 2.2 provides clear guidelines for marking holders and targets, resulting in heterogeneous and sometimes inconsistent annotations.

[20] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

We train two separate classifiers for holder and target expansion if there is enough training data, otherwise a joint classifier is used. Optionally, the system can merge multiple holder (target) spans for the same expression by adding missing tokens (typically, punctuation and 'and' tokens), so that a unique holder (target) span is extracted.

3.4 Polarity classification

The polarity module aims at computing the sentiment value from the extracted opinion expressions (see Section 3.2). This module is based on a model trained by using the Blitzer dataset[21] combined with information contained in SenticNet [3]. For each concept contained in SenticNet, the model contains a fuzzy membership function [19] describing the polarity of the concept and the uncertainty associated with it.

Model Construction In a preliminary learning phase an estimation of the polarity of each concept is inferred by analyzing explicit information provided by the training set. This phase allows to define the preliminary fuzzy membership functions associated with each concept. Such a value is computed as

$$\text{polarity}^{(LP)}(C_i) = \frac{k_{C_i}}{T_{C_i}} \in [-1, 1] \qquad \forall i = 1, \ldots, n,$$

where LP means "Learning Phase", C is the concept taken into account, n is the number of concepts contained in the model, k_{C_i} is the arithmetic sum of the polarities observed for concept C_i in the training set, and T_{C_i} is the number of instances of the training set in which concept C_i occurs. The shape of the fuzzy membership function generated during this phase is a triangle with the top vertex in the coordinates $(x, 1)$, where $x = \text{polarity}^{(0)}(C_i)$ and with the two bottom vertexes in the coordinates $(-1, 0)$ and $(1, 0)$ respectively. The rationale is that while we have one point (x) in which we have full confidence, our uncertainty covers the entire space because we do not have any information concerning the remaining polarity values.

After this, we compared the value computed through the training set with the one defined in the SenticNet ontology. This comparison shapes the fuzzy membership function of each term in the following way (see Figure 1):

$$\begin{aligned}
a &= \min\{\text{polarity}_i^{(LP)}(C), \text{polarity}_i^{(SN)}(C)\}, \\
b &= \max\{\text{polarity}_i^{(LP)}(C), \text{polarity}_i^{(SN)}(C)\}, \\
c &= \max\{a - ((b - a)/2)\}, \\
d &= \min\{b + ((b - a)/2)\}.
\end{aligned}$$

where SN refers to the polarity contained in SenticNet. Figure 1 shows a picture about an example of the final fuzzy trapezoid.

[21] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

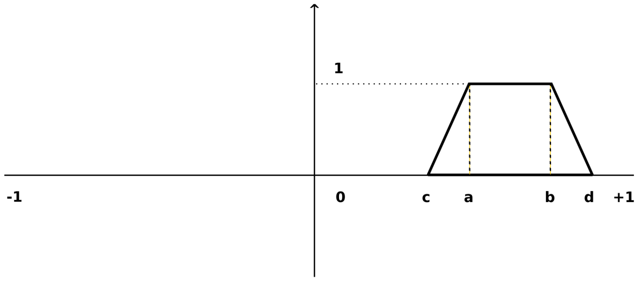

Fig. 1. The fuzzy trapezoid generated after the comparison between the polarity computed during the preliminary learning phase and the one contained in SenticNet.

Opinion Polarity Computation For each SenticNet concept identified in the opinion text, the correspondent fuzzy polarity is extracted from the model. The fuzzy polarities of different concepts are then aggregated by a fuzzy averaging operator obtained by applying the extension principle [20] in order to compute fuzzy polarities for complex entities, like texts, which consist of a number of concepts and thus derive, so to speak, their polarity from them. Details about how the set of membership functions are aggregated can be found in [7].

The result of the polarity aggregation phase is a fuzzy polarity, whose membership function reflects the uncertainty of the available estimate obtained by the system. Therefore, for extracting a crisp polarity value a defuzzification method, consisting in the conversion of a fuzzy quantity into a precise quantity, is needed. At least seven methods in the literature are popular for defuzzifying fuzzy outputs [9], which are appropriate for different application contexts. The *centroid method* is the most prominent and physically appealing of all the defuzzification methods. It results in a crisp value

$$y^* = \frac{\int y\mu_R(y)dy}{\int \mu_R(y)dy},$$

where the integration can be replaced by summation in discrete cases. This method is the one that we used for computing the value of the opinion polarity.

Finally, in the RAID system polarity is considered positive when $y^* > 0.2$, negative when $y^* < -0.2$, neutral otherwise.

4 Evaluation

We evaluate RAID on the MPQA, NTOA and DSRC datasets described in Section 2.2. We divided their documents into training and test sets according to a 75/25 ratio, except for the DSRC dataset where we reused the splitting given by the ESWC2015-CLSA challenge organizers. Then, we applied the RAID pipeline to extract opinion expressions, polarities, holders and (with the exception of the MPQA dataset) targets .

To compare the extracted expression, holder and target spans we use the state-of-the-art *intersection-based* precision and recall measures defined in [10]. Due to space reasons, we refer the reader to [10] for a detailed definition of these measures, only mentioning here that intersection-based measures evaluate extracted spans (of expressions, holder and targets) by giving them a reward proportional to the number of their tokens that intersect the ones of gold spans. This contrasts with *exact* measures, which require an exact match between extracted and gold spans, and *overlap-based* measures, which consider an extracted span as correctly marked if it overlaps with just one token of the gold standard (thus unfairly favoring longer extracted spans). Note that extracted and gold holder (target) spans are compared only when the corresponding expressions can be successfully matched, so to give credit only to holders (targets) of correctly extracted opinion expressions. Polarities are instead evaluated by comparing extracted and gold expressions that have been tagged (in gold and extracted data) with the same polarity, so that, e.g., the polarity precision measure corresponds to the amount of extracted opinion expressions whose span and polarity match the ones of gold expressions.

Table 2 shows RAID performance on the datasets described in Section 2.2 using the intersection-based measures. Concerning the identification of opinion expressions, the results show how the size of the dataset (and thus of training data) can result in different performances. The MPQA dataset, where RAID gets the best scores, is the biggest one, and this results in a high recall (0.501). On the contrary, NTOA is very small, that is the system has to be trained on a smaller set of expressions, resulting in high precision (0.819) and low recall (0.333). Finally, the DSRC dataset has some inconsistencies in the annotation of expressions, resulting in low scores overall. Similar considerations can be drawn for holder and target identification, noting that holder scores are generally (artificially) better for datasets such as DSRC whose opinion frames contain only few holder spans, as RAID successfully (and correctly) learns not to extract them.

To conclude the section, we report in Table 3 the performances of RAID on the DSRC dataset using the ESWC2015-CLSA evaluation measures, which are a form of exact precision-recall measures. In this setting, spans like “the teacher” and “teacher” are considered completely different, resulting in a decrease of precision and recall.

Table 2. RAID evaluation where precision (p), recall (r) and F_1-measure (f) are calculated for each dataset of Section 2.2. According to [10], polarity is considered wrong when the extracted expression cannot be matched to a gold expression. Accuracy for polarity detection without this limitation is 74.39 for MPQA, 83.67 for NTOA and 80.14 for DSRC.

	MPQA			NTOA			DSRC		
	p	r	f	p	r	f	p	r	f
expression	0.671	0.501	0.573	0.819	0.333	0.473	0.459	0.276	0.344
holder	0.584	0.584	0.584	0.710	0.647	0.677	0.952	0.952	0.952
target	-	-	-	0.416	0.431	0.424	0.596	0.588	0.592
polarity	0.419	0.305	0.353	0.620	0.283	0.389	0.321	0.195	0.243

Table 3. RAID evaluation using the ESWC2015-CLSA dataset and evaluation metric. Overall scores are the average of respective scores for expression, holder, target and polarity extraction.

	expression	holder	target	polarity	overall
precision	0.261	0.839	0.310	0.189	0.340
recall	0.416	0.964	0.452	0.302	0.534
F-value	0.321	0.897	0.368	0.232	0.455

5 Conclusions and future work

In this work we have presented RAID, a supervised tool for extracting opinion frames from texts. RAID first extracts the opinion expressions inside a sentence and, for each expression, it identifies the associated holder and target (if any) and assigns a positive / negative / neutral polarity to the opinion. RAID has been evaluated on three different datasets and its open-source code and a working demo are publicly available online.

We plan to improve RAID opinion extraction algorithm in the future, as well as the quality of the data used for training and evaluation of RAID. The latter aspect is motivated by the heterogeneous performances shown by RAID on different datasets, which have two explanations. First, the annotation guidelines differ among datasets, making difficult for a system to perform well on all datasets. For instance, the sentence 'Cameron called the crisis in Algeria a difficult, dangerous and potentially very bad situation' contains 5 different expression frames in NTOA, because its guidelines ask for splitting expression (as well as holder and target) spans when a conjunction ('and', 'or', . . .) is found. On the contrary, the MPQA dataset would consider them as a single opinion expression, unless different polarity values are involved. Second, datasets differ also regarding the completeness of their opinion annotations: while every opinion expression is guaranteed to be annotated in the MPQA dataset, this does not happen for DSRC, which includes many sentences where only part of the opinions are annotated.

Since the performances of a system depend on the size, consistency and quality of the training dataset, we deem useful to merge different opinion datasets into a single dataset with consistent annotations. This is something we would like to investigate in the future, so to increase the performances of RAID and other opinion extraction tools.

References

1. Mariana S C Almeida, Helena Figueira, and Pedro Mendes. Aligning Opinions : Cross-Lingual Opinion Mining with Dependencies. 2012.
2. Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2216–2220, 2010.
3. Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In *AAAI*, pages 1515–1521, 2014.
4. Erik Cambria, Bjö Bjorn Schuller, Yunqing Xia, and Catherine Havasi. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

5. Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 431–439, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
6. Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online Passive-Aggressive Algorithms. 7:551–585, 2003.
7. Mauro Dragoni, Andrea G.B. Tettamanzi, and Célia da Costa Pereira. Propagating and aggregating fuzzy polarities for concept-level sentiment analysis. *Cognitive Computation*, 7(2):186–197, 2015.
8. Aldo Gangemi, Valentina Presutti, and Diego Reforgiato Recupero. Frame-based detection of opinion holders and topics: A model and a tool. *IEEE Computational Intelligence Magazine*, 9(1):20–30, 2014.
9. H. Hellendoorn and C. Thomas. Defuzzification in fuzzy controllers. *Intelligent and Fuzzy Systems*, 1:109–123, 1993.
10. Richard Johansson and Alessandro Moschitti. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3):473–509, 2013.
11. Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, pages 1–8, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
12. Josef Ruppenhofer, Swapna Somasundaran, and Janyce Wiebe. Finding the sources and targets of subjective expressions. *The Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, (2), 2008.
13. Veselin Stoyanov and Claire Cardie. Annotating topics of opinions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.
14. Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 575–584, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
15. Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.
16. Michael Wiegand and Dietrich Klakow. Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 795–803, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
17. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
18. Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
19. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
20. Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - i. *Inf. Sci.*, 8(3):199–249, 1975.
21. Lei Zhang and Bing Liu. Aspect and entity extraction for opinion mining. *Data Mining and Knowledge Discovery for Big Data*, pages 1–40, 2014.