Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI),
Barcelona, Spain, July 2011, pp. 1758--1763.

Learning Cause Identifiers from Annotator Rationales

Muhammad Arshad Ul Abedin and **Vincent Ng** and **Latifur Rahman Khan**
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75080-3021
{arshad,vince,lkhan}@utdallas.edu

Abstract

In the aviation safety research domain, cause identification refers to the task of identifying the possible causes responsible for the incident described in an aviation safety incident report. This task presents a number of challenges, including the scarcity of labeled data and the difficulties in finding the relevant portions of the text. We investigate the use of annotator rationales to overcome these challenges, proposing several new ways of utilizing rationales and showing that through judicious use of the rationales, it is possible to achieve significant improvement over a unigram SVM baseline.

1 Introduction

The Aviation Safety Reporting System (ASRS) was established by NASA in 1976 to collect voluntarily submitted reports about aviation safety incidents written by flight crews, attendants, controllers and other related parties. Each report contains a *free text narrative* that describes, among other things, the cause of the incident. Knowing the causes of the incidents can play a significant role in developing aviation safety measures. As a result, we seek to automatically identify the causes of the incident described in a report narrative from a set of 14 possible causes, or *shaping factors*, identified by the NASA researchers [Posse *et al.*, 2005]. Since several factors may have caused an incident, cause identification is a multi-class, multi-labeled text classification problem.

This under-studied problem presents two major challenges to NLP researchers. One is the scarcity of labeled data: The original ASRS reports were not labeled with the shaping factors, and only a small part of it has so far been manually annotated, which makes it difficult for a supervised learning algorithm to generate an accurate model for all the shaping factors. Another is that the narratives chiefly describe the *incidents* that are occurring, while parts of the reports discussing the actual causes may be very small, which means the learner has to be careful as to which features it chooses as useful. This gives rise to two challenging questions: (1) how to solve the paucity of labeled data problem, and (2) how to enable the learner to make better use of the small pieces of clues lying in the text. One way to solve the data paucity problem is, of course, to annotate more data, but obtaining human-annotated data could be expensive. Interestingly, using the *annotator rationales* may provide an answer to these two questions.

An annotator rationale, as used by Zaidan *et al.* [2007], is the fragment of text that motivated an annotator to assign a particular label to a document. In their work on classifying the sentiment expressed in movie reviews as positive or negative, they generate additional training instances by removing rationales from documents. Since these pseudo-examples, known as *contrast examples*, lack information that the annotators thought as important, the learning algorithm should be less confident about the label of these *weaker* instances. A learner that successfully learns this difference in confidence assigns a higher importance to the pieces of text that are present only in the original instances. Thus the contrast examples help the learning algorithm both by providing indication to which parts of the documents are important and by increasing the number of available training instances. However, their approach explores only one way such rationales can be utilized.

We propose two ways to utilize the rationales for cause identification in a supervised learning framework. Specifically, the rationales are used to generate (1) new features and (2) a new type of pseudo-examples for the learner. This new type of pseudo-examples, called the *residue examples*, complements Zaidan *et al.*'s (2007) contrast examples in that they contain only the rationales present in a document. We motivate the use of residue examples in Section 4.

Experimental results on a set of manually annotated ASRS reports suggest that rationales are indeed helpful for cause identification. Using rationales as features in addition to generating both types of pseudo-examples, we find an improvement of over 7% in absolute F-score compared to a unigram SVM baseline that does not employ rationales.

The rest of the paper is organized as follows. In Section 2 we discuss the ASRS dataset. Section 3 describes Zaidan *et al.*'s [2007] framework for exploiting annotator rationales. We discuss and evaluate our own approach to utilizing the rationales in the cause identification problem in Sections 4 and 5, respectively. Finally, we perform a detailed error analysis in Section 6 and conclude in Section 7.

2 Dataset

The data set used here is based on the ASRS data set previously created by us [Abedin *et al.*, 2010]. It contains 1333 reports labeled with shaping factors[1] (see the column "Shaping Factor" in Table 1 for the list of the 14 shapers[2]). These reports are divided into three subsets: a training set containing 233 reports, a development set containing 100 reports and a test set containing 1000 reports. Since we need reports annotated with rationales, we decided to generate some additional training data, choosing 1000 unlabeled reports randomly from those we provided in Abedin *et al.* [2010] and adding them to the training set. Then we annotated the newly-added reports with shaping factors, and identified annotation rationales for all the reports in the training set. Below, we discuss the procedures in annotating a report with shaping factors (Section 2.1) and rationales (Section 2.2).

2.1 Annotation With Shaping Factors

We followed the same annotation procedure described in Abedin *et al.* [2010] to assign shaping factor labels to the 1000 unlabeled reports that we added to the training set. One author (A1) and one student worker (A2) independently annotated 100 randomly chosen unlabeled reports, using the descriptions of the shaping factors in Posse *et al.* [2005] and Ferryman *et al.* [2006]. The inter-annotator agreement, calculated using the Krippendorff's (2004) α statistics as described by Artstein and Poesio [2008] with the MASI scoring metric [Passonneau, 2004], was found to be 0.82. The annotators then resolved each disagreement through discussion. Given the high agreement rate, we had A1 annotate the remaining reports. Thus, at the end of the annotation process, we had a training set with 1233 reports labeled with shaping factors.

2.2 Annotation With Rationales

The same two annotators, A1 and A2, went through the same 100 reports and answered the question below for each report:

> For each shaping factor identified for the incident described in the report, is there a fragment of text that is indicative of that shaping factor?

If so, we provide A1 and A2 with the same instructions that Zaidan *et al.* had for their annotators for marking up the rationales (see Section 4.1 of their paper). Briefly, the annotators are asked to "do their best to mark enough rationales to provide convincing support for the class of interest", but are not expected to "go out of their way to mark everything".

The inter-annotator agreement for the rationales were calculated in the same manner as Zaidan *et al.* [2007], where two rationales are considered overlapping if they have at least one word in common. In the 100 reports, A1 annotated 257 rationales, 54.5% of which overlap with A2's rationales, and A2 identified 182 rationales, with a 76.9% overlap. Statistics for all the classes are shown in Table 2. A1 then proceeded to annotate the rationales for the remaining 1133 reports in training set, resulting in a total of 5482 rationales over the 1233 training reports, for an average of 4.4 rationales per document. To get a better sense of these rationales, we list the five most frequently-occurring rationales annotated for each shaping factor in Table 1. Since rationales are meant to improve classifier *acquisition*, the test reports do not require (and therefore do not contain) any rationale annotations.

[1] See http://www.utdallas.edu/∼maa056000/asrs.html

[2] Space limitations preclude the inclusion of the definitions of these shaping factors. See Abedin *et al.* [2010] for details.

	Annotator A1		Annotator A2	
Class	Rationales	Overlap with A2	Rationales	Overlap with A1
1	12	75.0%	9	100.0%
2	22	77.3%	17	100.0%
3	6	50.0%	3	100.0%
4	8	87.5%	9	77.8%
5	0	0.0%	0	0.0%
6	26	42.3%	15	73.3%
7	42	50.0%	34	61.8%
8	16	75.0%	12	100.0%
9	9	44.4%	9	44.4%
10	10	100.0%	11	90.9%
11	25	32.0%	12	66.7%
12	64	37.5%	28	85.7%
13	11	72.7%	10	80.0%
14	6	100.0%	13	46.2%
Total	257	54.5%	182	76.9%

Table 2: Inter-annotator agreements for rationales.

3 Rationales in Sentiment Classification

In this section, we describe Zaidan *et al.*'s [2007] approach to training an SVM with annotator rationales for classifying the sentiment of a movie review as positive or negative.

Let $\mathbf{x}_i$ be the vector representation of document R_i. Given the rationale annotations on a positive example $\mathbf{x}_i$ for the SVM learner, Zaidan *et al.* construct one or more not-so-positive *contrast* examples $\mathbf{v}_{ij}$. Specifically, they create $\mathbf{v}_{ij}$ by removing rationale r_{ij} from R_i. Since $\mathbf{v}_{ij}$ lacks evidence that an annotator found relevant for the classification task, the correct SVM model should be less confident of a positive classification on $\mathbf{v}_{ij}$.

This idea can be implemented by imposing additional constraints on the correct SVM model, which can be defined in terms of a weight vector $\mathbf{w}$. Recall that the usual SVM constraint on positive example $\mathbf{x}_i$ is $\mathbf{w} \cdot \mathbf{x}_i \geq 1$, which ensures that $\mathbf{x}_i$ is on the positive side of the hyperplane. In addition to these usual constraints, we desire that for each j, $\mathbf{w} \cdot \mathbf{x}_i - \mathbf{w} \cdot \mathbf{v}_{ij} \geq \mu$, where $\mu \geq 0$. Intuitively, this constraint specifies that $\mathbf{x}_i$ should be classified as more positive than $\mathbf{v}_{ij}$ by a margin of at least μ.

Let us define the annotator rationale framework more formally. Recall that in a standard soft-margin SVM, the goal is to find $\mathbf{w}$ and ξ to minimize

$$\frac{1}{2}|\mathbf{w}|^2 + C\sum_i \xi_i$$

subject to

$$\forall i : c_i \mathbf{w} \cdot \mathbf{x}_i \geq 1 - \xi_i,$$
$$\xi_i > 0,$$

Id	Shaping Factor	Rationales
1	**Attitude**	attitude (24), complacency (13), complacent (6), playing a game (2), overconfident (2)
2	**Communication Environment**	noise (28), no response (18), did not hear (14), static (12), congestion (12)
3	**Duty Cycle**	last leg (7), last night (4), reduced rest (3), longitude duty days (2), longitude duty day (2)
4	**Familiarity**	new (123), unfamiliar (16), aligned (9), not familiar (5), very familiar (4)
5	**Illusion**	bright lights (2), wall of white (1), black hole (1)
6	**Physical Environment**	weather (144), visibility (77), turbulence (51), clouds (43), winds (38)
7	**Physical Factors**	fatigue (25), tired (14), sick (9), fatigued (7), very tired (5)
8	**Preoccupation**	busy (78), attention (76), distraction (30), distracted (17), DISTRS (9)
9	**Pressure**	late (56), pressure (45), expedite (14), short time (12), rushed (12)
10	**Proficiency**	training (57), mistake (42), inadvertently (27), mistakes (18), forgotten (11)
11	**Resource Deficiency**	off (410), more (194), further (133), damage (85), warning (84)
12	**Taskload**	very busy (13), solo (13), extremely busy (9), many aircraft (8), single pilot (7)
13	**Unexpected**	surprised (14), suddenly (13), unexpected (7), unusual event (2), unknown to me (2)
14	**Other**	Resolution Advisory (59), confusion (58), confused (21), confusing (13), UNCLR (9)

Table 1: Five most frequently-occurring rationales associated with each shaping factor and their frequencies of occurrences.

where $\mathbf{x}_i$ is a training example; $c_i \in \{-1, 1\}$ is the class label of $\mathbf{x}_i$; ξ_i is a slack variable that allows $\mathbf{x}_i$ to be misclassified if necessary; and $C > 0$ is the misclassification penalty. To enable this standard soft-margin SVM to also learn from contrast examples, Zaidan *et al.* add *contrast* constraints:

$$\forall i, j : \mathbf{w} \cdot \mathbf{x}_i - \mathbf{w} \cdot \mathbf{v}_{ij} \geq \mu(1 - \xi_{ij}^{'}),$$

where $\mathbf{v}_{ij}$ is the contrast example created from R_i by removing rationale r_{ij}, and $\xi_{ij}^{'} \geq 0$ is the slack variable associated with $\mathbf{v}_{ij}$. These new slack variables should have their own misclassification cost, so a new term is added to the SVM objective function, which becomes:

$$\frac{1}{2}|\mathbf{w}|^2 + C \sum_i \xi_i + C^{'} \sum_{i,j} \xi_{ij}^{'}$$

Here, $C^{'}$ is the misclassification cost associated with the new slack variables $\xi_{ij}^{'}$.

4 Rationales in Cause Identification

Before describing how we employ rationales for cause identification, recall that cause identification is a 14-class classification task. To employ Zaidan *et al.*'s annotator rationale framework, which relies on SVM training and hence assumes only positive and negative examples, we recast cause identification as a set of 14 binary classification problems, one for predicting each shaper. More specifically, in the binary classification problem for predicting shaper s_i, we create one training example $\mathbf{x}_i$ from each document in the training set, labeling it as positive if the document has s_i as one of its labels, and negative otherwise, In essence, we are adopting a *one-versus-all* scheme for creating original training instances.

Next, we describe how we can employ rationales for cause identification. We use rationales to provide additional *features* and additional *training examples*. Using rationales to create additional features is fairly straightforward. It involves (1) constructing a special lexicon that contains all the rationales in the training data, (2) creating one feature for each rationale in the lexicon, and (3) augmenting the feature set with these rationale features. The value of a rationale feature is 1 if the string representing the rationale appears in the given report and 0 otherwise.

We use rationales to provide two types of additional training examples: *contrast* examples and *residue* examples. To use rationales to generate contrast examples, we follow Zaidan *et al.*'s approach, as described in Section 3. However, there is a caveat. In Zaidan *et al.*'s framework, rationales are used to generate both positive *and* negative examples, since rationales were collected from both positive and negative reviews. On the other hand, in cause identification, since each report can be labeled with more than one shaping factor, the presence of a rationale for shaper s_j in a report does not provide evidence that the report should not be labeled with shaper s_i, where $i \neq j$. Rather, it is the *absence* of rationales for s_i rather than the presence of rationales for s_j that indicates that the report should not be labeled with s_i. Hence, when recasting cause identification as binary classification tasks, the absence of rationales for the negative examples implies that we will only generate positively-labeled contrast examples from the rationales. In particular, these positively-labeled contrast examples are generated to train each binary classifier in the same way as in Zaidan *et al.*'s framework.

Now, recall that we propose to a new type of training examples from the rationales: *residue* examples. Hence, Zaidan *et al.*'s framework needs to be modified so that the SVM learner can take into account the residue examples. Before describing the modifications, let us motivate the use of residue examples. Recall from Section 2.2 that the rationale annotators were not expected to "go out of their way" to identify all the rationales in a document. Hence, any amount of rationale annotation is unlikely to mark *all* the relevant portions of the text, and thus, the text outside of the rationales may also contain some relevant portions. Thus, we propose residue examples, which are examples that contain only the rationales. Specifically, for each rationale r_{ij} appearing in training example $\mathbf{x}_i$, we generate one (positively-labeled) residue example, $\mathbf{r}_{ij}$, which contains only the features extracted from r_{ij}. In a sense, the contrast examples help the learner understand the importance of the rationale, whereas the residue examples aid in learning the importance of the *rest* of the text.

As with contrast examples, residue examples intuitively

Expt	Feature Set	P	R	F
SVM Baseline				
1	Unigrams	50.8	36.2	42.2
SVM with Rationales as Features				
2	Rationales	54.8	34.8	42.5
3	Unigrams, Rationales	56.0	38.8	45.8

Table 3: Results of the baseline and the SVM approach with rationales as features.

contain less relevant information for classification than the original documents, and hence the SVM model should also be less sure of their class values than the original documents. To capture this intuition in Zaidan *et al.*'s framework, we add *residue* constraints:

$$\forall i, j : \mathbf{w} \cdot \mathbf{x}_i - \mathbf{w} \cdot \mathbf{r}_{ij} \geq \mu(1 - \xi_{ij}^{''}),$$

where $\mathbf{r}_{ij}$ is the residue example created from $\mathbf{x}_{ij}$ by retaining only rationale r_{ij}, $\mu \geq 0$ is the size of the minimum separation between original and residue examples, and $\xi_{ij}^{''} \geq 0$ is the slack variable associated with $\mathbf{r}_{ij}$. These slack variables should have their own misclassification cost, so a new term is added to Zaidan *et al.*'s objective function, which becomes:

$$\frac{1}{2}|\mathbf{w}|^2 + C\sum_i \xi_i + C^{'}\sum_{i,j} \xi_{ij}^{'} + C^{''}\sum_{i,j} \xi_{ij}^{''}$$

Here, $C^{''}$ is the misclassification cost associated with $\xi_{ij}^{''}$.

5 Evaluation

In this section, we evaluate the usefulness of rationales for cause identification in a supervised learning framework.

5.1 Baseline Results

Following Zaidan *et al.* [2007], our baseline uses SVM^{light} [Joachims, 1999] with a linear kernel to train one-versus-all classifiers in conjunction with a feature set comprising only unigrams on the 1233 training reports without pseudo-examples. Numbers, punctuations, stopwords, and tokens appearing less than 3 times in the training set are excluded from the feature set. The feature values are all boolean: if the token appears in the report, its value is 1 and 0 otherwise. The only SVM learning parameter in this case is C, the misclassification penalty, which was selected as the one that achieved the best performance on the 100-report development set (see Section 2). Results are expressed in terms of precision (P) and recall (R), both of which are micro-averaged over those of the 14 binary classifiers, as well as F-score (F). As we can see from row 1 of Table 3, the baseline achieves an F-score of 42.2 on the 1000-document test set.

5.2 Results Using Rationales

Parameter tuning. We tune the cost parameters C, C' and C'', and the pseudo-example parameter μ to maximize the F-score of the SVM model on the development set. Due to the large number of parameters, we use the *design of experiment*-based search method for parameter tuning proposed by Staelin [2002] to find the parameter combination that gives the best F-score on the development set. Specifically, we specify the search space of each parameter, and the algorithm specifies several points of evaluation. At each point, the SVM models are learned from the training data using the parameter values at that point, and the learned models are used to classify the development test set instances. The point at which the best performance (F-score) is observed is then selected as the new center of search and the search space ranges are halved. This search process is repeated 5 times. The models built using the best parameter combination after 5 iterations are then applied to the test set.

Using rationales as features. Results of using the rationales as features are shown in rows 2 and 3 of Table 3. Note that the frequency threshold of 3 mentioned earlier was employed to filter the unigram features but not the rationale features. When only the rationales are present in the feature set, the SVM model performs marginally better than the baseline, achieving an F-score of 42.5. However, when rationales are used as *additional* features to augment the unigram-based feature set, the F-score increases considerably to 45.8. This shows that the rationales are useful when used as features.

Using rationales as pseudo-examples. Results of using the rationales to generate pseudo-examples are shown in Table 4. As we can see, using only contrast examples (row 4) does not improve much upon the SVM baseline. Interestingly, using our proposed residue examples (row 5) improves the SVM baseline considerably to F=49.0. When both types of pseudo-examples are applied (row 6), F-score drops to 47.5, which still performs well above the baseline, however.

Next, we repeat the above experiments, but augment the feature set with rationales. Comparing the three pairs of experiments (rows 4 and 7; rows 5 and 8; rows 6 and 9), we can see that using rationales both as features and as contrast examples consistently yields better performance than using rationales only as contrast examples. Hence, rationale features are useful for cause identification in the presence of contrast examples.

Finally, when rationales are used as features, neither residue examples nor contrast examples are more effective than the other in improving the SVM model (rows 7 and 8), but employing both types of pseudo-examples enables the model to achieve substantially better performance (row 9).

Using rationales as pseudo-negatives. So far we have only used rationales to provide pseudo positive examples. A natural question is: while we do not have rationales that explain why a document should not be labeled with a particular shaper, is it still possible to generate pseudo negative examples? We experiment with a simple idea for generating pseudo-negatives: to train the binary classifier for predicting shaper s_i, we create negatively-labeled contrast examples from the rationale annotations in the negative examples in the same way as we create positively-labeled examples from the rationales in the positive examples. The next question, of course, is: since the rationales in the negative examples are not necessarily indicators that a document should not be labeled as s_i, will these negative contrast examples be useful? To answer this question, we repeat the experiments in Table 4,

Expt	Feature Set	Pseudo-Instances	P	R	F
4	Unigrams	Contrast	49.9	38.1	43.2
5	Unigrams	Residue	36.9	72.8	49.0
6	Unigrams	Contrast, Residue	36.4	68.5	47.5
7	Unigrams, rationales	Contrast	52.2	40.7	45.7
8	Unigrams, rationales	Residue	55.9	38.8	45.8
9	Unigrams, rationales	Contrast, Residue	39.7	66.0	**49.5**

Table 4: Results of approaches using pseudo-examples.

but augment the training set in each experiment with negative contrast examples. Our preliminary results indicate that additional employing pseudo-negatives yields an improvement of 1.0–2.6 in F-score, but further experiments are needed to precisely determine their usefulness.

Generating pseudo-examples from multiple rationales. So far each pseudo-example is generated from exactly one rationale. It is conceivable that a contrast example can be generated by removing multiple rationales and a residue example can be generated by retaining multiple rationales. Adding this flexibility to the generation of pseudo-examples may result in an explosion in the number of pseudo-examples, however. As a result, we limit the number of pseudo-examples that can be generated for each training report. Specifically, to generate pseudo-examples for a report with r rationales, we (1) choose l such that $\sum_l \binom{r}{l} \leq 50$, and (2) remove/retain rationale combinations of sizes $1, 2, \ldots, l$ from the report when generating contrast/residue examples. Adding this flexibility to the generation of positive and negative contrast and residue examples, we achieve an F-score of 52.9 in an experiment where both unigrams and rationales are employed as features. This preliminary result indicates that it may be beneficial to generate pseudo-examples from multiple rationales.

6 Error Analysis

To better understand why the systems fail to correctly classify a number of test instances, we analyze the errors made by the best-performing system (i.e., the system that yielded an F-score of 52.9) on a set of 100 reports chosen randomly from the test set. We investigate two types of errors, namely *false positives*, in which the system labels a report as positive but the annotator does not, and *false negatives*, in which the annotator labels a report as positive but the system does not.

6.1 Analysis Method

Analyzing the errors made by an SVM learner is a rather daunting task since the actual reason of why a particular instance is classified as positive or negative is buried under the vectors representing the separating hyperplane and the document being classified. The SVM learner takes the dot product of these two vectors (and adds the bias term if biased hyperplanes are being used) to find the distance of the instance from the separating hyperplane, and if the distance is greater than zero then it is classified as positive, otherwise it is labeled negative. Hence, to analyze the causes of the false positives and the false negatives, we need to look at the actual features values present in the document vector and their weights in the hyperplane vector. However, even with a subset of 100 reports, this means looking at more than 60,000 feature values, and thus we confine ourselves to looking at only the highest contributing features. In other words, for the false positives, we look at the feature with the highest positive contribution to the distance, and for the false negatives, we look at the one making the biggest negative contribution.

False Positives	62	Percentage
Concept present but not contributing to incident	21	33.87%
Wrong context	17	27.42%
Bad feature	12	19.35%
Too general feature	7	11.29%
Wrong word sense	3	4.84%
Hypothetical context	1	1.61%
Ambiguous feature	1	1.61%

Table 5: Error analysis findings for false positives.

6.2 False Positives

We analyzed the 62 false positive errors made by the system and discovered several different reasons why the top positive contributing features actually misled the learner. These reasons are summarized in Table 5 and discussed in detail below.

Concept present but not contributing to incident. In several cases, the top positive contributing feature *is* in fact relevant to the shaping factor, but in the specific report, it is not contributing to the incident in the opinion of the annotator. For example, the term "busy" is a feature that is relevant to the shaping factor *Preoccupation*, but not every report in which this term appears has *Preoccupation* as a shaper. For example, Report#230334 contains the sentence "CLBING to cruise altitude we were cleared to FL230 by a busy controller." Even though the controller was evidently busy, that did not contribute to the occurrence of the incident.

Wrong context. In this category of errors, the feature is in fact relevant to the shaping factor, but it appears in such a context that does not imply the shaping factor. For example, the feature "damage" related to the shaping factor Resource Deficiency appears in Report#389873 in the following context: "minimal damage." In this context, the word appears as an *effect* of what happened, not as a *cause*. The SVM learner is unable to make this distinction, and the feature ends up contributing heavily to the positiveness of the document.

Bad feature. There are several cases in which the feature is not intuitively relevant to the shaping factor, but were assigned high weights by the learner merely because they happened to occur frequently enough in the positive examples in the training set. For example, the feature "ability" is the top positive contributor for the shaping factor *Attitude* in Report#101846, as in the sentence "aircraft performance charts and PDCS affirmed ability to do so", but it is clear that the word "ability" has nothing to do with an attitude problem.

Too general feature. There are some words that are related to the shaping factor in a very general manner, but because of this generality, may appear in any report without actually indicating the shaping factor. Similar to the bad features described above, the learner assigns them a high weight because of their frequent occurrence in the positive reports. For example, the feature "weather" is related to the shaping factor *Physical Environment* in a general sense since this shaping factor is chiefly related to the weather elements hampering a pilot, but in the context of Report#325010, where it appears in the sentence "the weather was clear with good visibility for the los angeles area", it is merely appearing to describe the weather conditions.

Wrong word sense. In this case the one sense of the feature is relevant to the shaping factor, but it appears in a different sense in the falsely positive report. For example, Report#642907 is labeled as a positive example for class Attitude, and the feature with the highest positive contribution is the word "attitude". This word is apparently a good indicator for this class as this has also been identified by the annotators as a rationale for this class. However, this is true only as long as it means the mental state of someone. In this particular report, this word appears in the following context: "I became slightly disoriented and got into a dangerous attitude." In this context the word "attitude" is actually used to mean the orientation of the aircraft and not the mental state of the pilot. The SVM learner understandably cannot make the difference between these two meanings and this feature makes a high contribution to the false positive label of the document.

Others. In one case, the word "tired", a good feature for the shaper *Physical Factors*, appears in Report#534432 in a hypothetical context: "I feel this could happen to a pilot especially if he was single pilot; behind on the approach; a little rusty; tired; etc. ...". In another case, the top positive contributor is the feature "fatigue", which is relevant to two shapers, *Duty Cycle* and *Physical Factors*. Thus the report gets labeled positive for *Duty Cycle*, resulting in a false positive.

6.3 False Negatives

In the analysis of the false negatives, we look at the highest *negative* contributing feature for the reports falsely labeled as negatives. However, in this case it is harder to identify whether the top contributing feature is a good negative feature or not. For example, it is easier to judge whether a given feature is relevant to a given shaping factor, but how do we judge whether a feature would be a good indicator for the shaping factor *not* being present?

This exposes one interesting property of our problem of cause identification. When a shaping factor *is* present, it reflects in the report as such, and by utilizing the clues in the document, it is possible to identify its presence to a reasonable degree. However, when the shaping factor is *not* present, these is rarely any clue. In other words, the person writing the report usually does not discuss the shaping factors that were not present. When we look at the top negative contributing features for the false negatives, we find that the SVM learner facing this issue. In some of the cases, it tries to solve the problem by assigning high negative weights to features important to other classes, while in other cases it simply focuses on random words that appear in the negative examples more than the positive ones. The first approach creates false negatives when that other shaping factor is also present in the test document (as we discuss in the next paragraph), while the second approach mostly selects bad features.

Feature associated with other labels of the document We take each feature and find the shaping factor for which the feature has the highest positive weight in the vector representation of the hyperplane. Then, when we look at the false positive example, we find the highest negative contributing feature, and look up its most relevant shaping factor. Then we see if this shaping factor is also present in the set of labels assigned to the report. In 36 of the 87 false positives we analyzed, we found that the feature is affiliated with one of the other shaping factors assigned to the report. Thus, since the features relevant to the other shaping factors have been assigned negative weights, their presence in the report makes them contribute negatively even though they are present because of the presence of other shaping factors in the report.

7 Conclusions

We have proposed to use annotator rationales for cause identification, suggesting two novel ways to use rationales — as features and for generating a new type of pseudo-examples, namely residue examples. Experimental results on a subset of the ASRS reports demonstrated the usefulness of rationales for cause identification. Overall, an SVM learner that exploits rationales substantially improves one that does not by 7.3% in F-score. Moreover, we believe that our detailed analysis of the errors made by our system can provide insights into the problem as well as directions for future research.

References

[Abedin *et al.*, 2010] M. Abedin, V. Ng, and L. Khan. Weakly supervised cause identification from aviation safety reports via semantic lexicon construction. *JAIR*.

[Artstein and Poesio, 2008] R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Comp. Linguistics*.

[Ferryman *et al.*, 2006] T. A. Ferryman, C. Posse, L. J. Rosenthal, A. N. Srivastava, and I. C. Statler. What happened, and why: Toward an understanding of human error based on automated analyses of incident reports Volume II. Technical Report NASA/TP–2006-213490, NASA.

[Joachims, 1999] T. Joachims. Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

[Passonneau, 2004] R. J. Passonneau. Computing reliability for coreference annotation. In *LREC*.

[Posse *et al.*, 2005] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke, and T. Ferryman. Extracting information from narratives: An application to aviation safety reports. In *2005 IEEE Aerospace Conference*.

[Staelin, 2002] C. Staelin. Parameter selection for support vector machines. Technical Report HPL-2002-354R1, HP Labs Israel.

[Zaidan *et al.*, 2007] Omar F. Zaidan, J. Eisner, and C. Piatko. Using "annotator rationales" to improve machine learning for text categorization. In *NAACL HLT*.