Wordplay: An Examination of Semantic Approaches to Classify Safety Reports

Shawn R. Wolfe.[1]
NASA Ames Research Center, Moffett Field, CA, 94035

Aviation safety reports, such as those of the Aviation Safety Action Program (ASAP) and the Aviation Safety Reporting System (ASRS), provide a valuable record of safety incidents for industry analysts. Unfortunately, the sheer quantity of safety reports often makes it difficult to identify recurring safety issues and their causes. In response, NASA has explored the use of text classification techniques to automatically identify safety issues from the text of safety reports. One common approach for text classification is to use a statistical model of word occurrences in each report (often referred to as a bag of words), and use these models to train a classifier. We performed an experiment to evaluate whether simple semantic approaches could improve classification results with a support vector machine (SVM) classifier by factoring semantic relationships of words into the statistical model. Counter to our intuition, most of these semantic enhancements did not improve classification results. One method of combining meaningful words did show an improvement over the standard approach, however, and is presented along with the results of our study.

I. Introduction

AVIATION safety reports can provide a valuable record of safety incidents. The Aviation Safety Action Program (ASAP) and the Aviation Safety Reporting System (ASRS) are two successful programs that systemize and standardize the reporting of safety incidents. These incident reports are stored in several archives and provide a means for industry analysts to identify and monitor recurring safety issues. The volume of such reports is staggering: over 600,000 reports have been submitted to ASRS[1] with more than 40,000 new reports submitted in 2005 alone[2]. This overwhelming quantity of information surpasses the ability of an analyst to review every report. In part, this is mitigated by the use of classification schemes to group reports into categories of potential interest, such as the type of incident or situation characteristics. This allows the analysts to quickly identify a subset of reports in the corpus that are relevant to the current study.

In reality, the challenge of reviewing each report remains, but it has been transferred from the analysis phase to a classification phase. This potentially reduces the cost by performing the classification only once per report, rather than once for each analyst, benefiting all analysts who use the archive. The task of classifying such a large quantity of reports remains daunting, however. Typically, a team of highly trained experts will review each report and arrive at a mutually agreeable set of classifications[1]. The requisite expertise is rare, and several teams of experts are needed to keep up with the incoming flow of documents. Coupled with the cost of manually encoding the reports, it is not surprising that no more than one in five submitted reports are entered into the ASRS database[3]. Since the majority of reports are unavailable for analysis, it is unlikely that the full benefit of ASRS is currently realized.

Any technology that decreases the difficulty of categorizing aviation safety reports could both increase the benefit and decrease the cost of managing a program such as ASRS. An effective tool that assists an expert in classifying incident reports would save effort; a system that automatically and reliably classifies a portion of the incident reports without expert guidance could have an even greater impact. In support of these goals, NASA has investigated the application of text classification technologies to classify aviation safety reports.

The most popular text classification techniques use domain-independent statistical models of the text and sophisticated machine learning algorithms to classify text. Improving upon these underlying algorithms remains an active area of research, but is outside the scope of our investigation. Rather, we sought to evaluate whether the introduction of domain knowledge could significantly improve the performance of a standard text classifier for aviation safety reports. Our hypothesis was that the lack of domain knowledge was negatively impacting

[1] Computer Scientist, Intelligent Systems Division, Mail Stop 269-2. Member, AIAA.

classification results, and even a modest amount of incorporated domain knowledge could improve performance. Unfortunately, domain knowledge is typically expensive to incorporate into a computer system and is not commonly a part of text classification (which instead relies on implied relationships derived through statistical analysis rather than an explicit representation of meaning). In this work, we only approximate the effect semantic information might have on classification results in order to measure its potential without incurring the full developmental cost.

II. Related Work

In our study, we represent each document in terms of weighted statistics of word occurrences, commonly referred to as a bag of words, and use a support vector machine (SVM) to classify the documents. The bag of words model and TF*IDF term weighting scheme was initially developed in the context of information retrieval[4,5]. Support vector machines were introduced by Vapnik as a general method for statistical learning; their applicability to text classification was argued by Joachims[6] and has since become a popular approach to text classification[7].

Automatic classification of aviation safety reports has been examined in previous studies. Decision trees were used to classify Southwest Airlines ASAP reports in a proof of concept study[8]. Srivastava et al. used SVMs to classify ASRS reports[9], and have since expanded on their earlier work with a system called Mariana[10]. Their effort has been a strong influence on our study; we have attempted to recreate their techniques[9] in our baseline when possible, but acknowledge that differences in implementation are inevitable.

Information extraction techniques have also been used to classify JetBlue and United Airlines safety reports by looking for key phrases in the text that are representative of a given incident type[11,12]. A similar approach was used to classify ASRS reports[13], which is relevant to our approach in that lists of related terms were also exploited to modify the bag of words. However, the phrases and term lists were engineered manually and applied globally, unlike our approach where all term lists were derived statistically, and different term lists were used for each incident class. A logic-based approach was also used to derive a concept lattice from the fixed field data of ASRS[14].

Statistically deriving relationships between terms has been explored previously within the context of aviation safety reports[15]. More generally, statistical techniques for dimensionality reduction such as Principal Component Analysis and Singular Value Decomposition are similar to our use of term lists[16]. Koller and Sahami developed a hierarchical method for classifying documents with very few words[17], a much more sophisticated approach than the simple one word classifier we used for comparison.

III. Aviation Safety Reporting System

We used a corpus consisting of 20696 ASRS reports as our dataset. Every report in the corpus had been classified previously, with most reports classified in more than one incident type. We limited ourselves to the forty most common incidents, which together occurred in all but 135 reports. Figure 1 shows the distribution of incident classes and number of incident classes per report, given our restricted set of incident classes. Metadata (such as phase of flight, time of day, etc.) were also encoded for each report but were not used in our study. Instead, we used only the narrative text of the reports, which vary from a single sentence to several paragraphs in length. Figure 2 shows a representative narrative[18]. ASRS reports contain a high number of domain specific terms, nonstandard abbreviations, and shorthand. The use of abbreviations is fairly regular but not entirely so; abbreviated and complete versions of the same word occur in the text (for example, ACFT and AIRCRAFT), and some abbreviations expand out into different words. These issues make the reports more challenging to work with, but are not unique problems: synonymy and polysemy are a normal part of language.

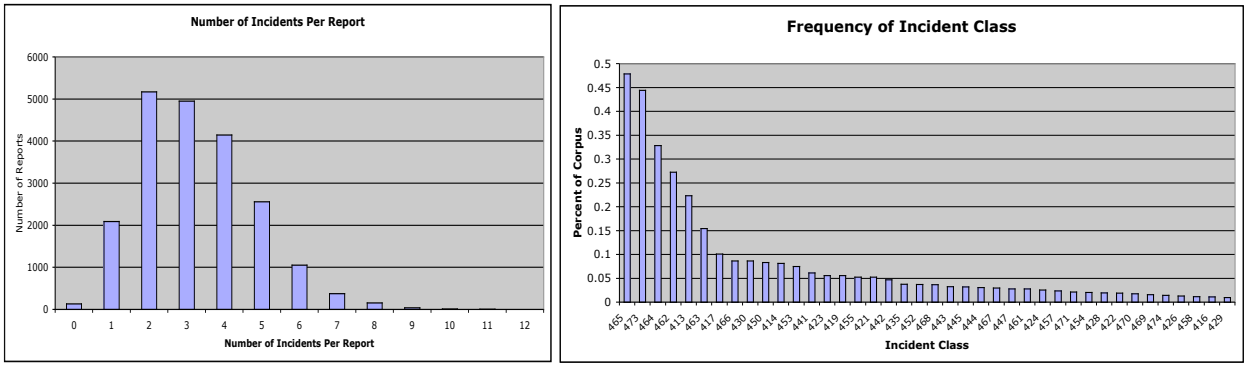

Figure 1. Distribution of incidents and incidents per report.

The assigned classifications in the corpus were not always consistent: in one extreme case, two reports with identical narratives had different classifications. There are several reasons for this. First, important differences may lie in the metadata of reports with similar narratives, leading to different classifications. Second, reports are not all from the same period of time, and the prevailing wisdom as to what constitutes a certain safety incident is likely to drift over time. Finally, even aviation experts do not always agree on the nature of a safety incident. To a degree, this is reduced by the requirement of consensus with the team, but different teams may still classify the same report differently. Regardless of the reason, the variation in classification is an additional challenge when learning from the ASRS corpus.

PAX ARRIVED BRU DEC/MON/04. HAD MALARIA ATTACK 3 WKS PRIOR IN SIERRA LEONE (SAW DOCTOR). IN FRA, REQUESTED WHOLE ROW FLT ON DEC/TUE/04 AND SAID HE WASN'T FULLY RECOVERED. GND STAFF ALERTED ME AND I ALERTED CAPT. HAD GND STAFF CALL DOCTOR WHO SAID HE WASN'T CONTAGIOUS AND COULD FLY. 2 1/2 HRS PRIOR TO LNDG, DISPATCH CONTACTED CAPT THAT PAX MAY BE CONTAGIOUS AND THAT PUBLIC HEALTH WOULD MEET FLT. ALSO ADVISED ALL PAX HAD TO REMAIN ON BOARD UNTIL CLRED BY PUBLIC HEALTH. PUBLIC HEALTH REFUSED TO MEET FLT. (WE WERE NOT ADVISED UNTIL LNDG.) FLT MET BY AGENT Y AND Z (MGRS, COMPANY CUSTOMS). SO, FOR 2 1/2 HRS, ENTIRE CREW THOUGHT WE MAY BE QUARANTINED. ADVISED GND STAFF TO KEEP US IN THE KNOW IN FUTURE.

Figure 2. Example narrative from ASRS report ACN: 642270

term	tf	idf	tf*idf	term	tf	idf	tf*idf	term	tf	idf	tf*idf	term	tf	idf	tf*idf
04	2	5.86	11.72	COULD	1	1.33	1.33	LEONE	1	9.83	9.83	SIERRA	1	6.74	6.74
1/2	2	4.93	9.86	CREW	1	1.92	1.92	LNDG	2	1.25	2.50	SO	1	1.40	1.40
2	2	1.04	2.08	CUSTOMS	1	5.33	5.33	MALARIA	1	9.83	9.83	STAFF	3	6.00	18.00
3	1	1.53	1.53	DEC	2	4.70	9.40	MAY	2	2.37	4.74	THAT	2	0.39	0.78
ADVISED	3	1.86	5.58	DISPATCH	1	2.97	2.97	ME	1	1.24	1.24	THE	1	0.08	0.08
AGENT	1	3.81	3.81	DOCTOR	2	5.12	10.24	MEET	2	4.02	8.04	THOUGHT	1	2.14	2.14
ALERTED	2	4.43	8.86	ENTIRE	1	3.50	3.50	MET	1	3.61	3.61	TO	4	0.03	0.12
ALL	1	1.34	1.34	FLT	4	5.06	20.24	MGRS	1	7.19	7.19	TUE	1	4.25	4.25
ALSO	1	1.51	1.51	FLY	1	2.38	2.38	MON	1	4.55	4.55	UNTIL	2	1.94	3.88
AND	5	0.06	0.30	FOR	1	0.37	0.37	NOT	1	0.45	0.45	US	1	0.98	0.98
ARRIVED	1	3.25	3.25	FRA	1	7.89	7.89	ON	2	0.20	0.40	WASN'T	2	0.45	0.90
ATTACK	1	5.48	5.48	FULLY	1	3.96	3.96	PAX	3	5.24	15.72	WE	2	0.46	0.92
BE	2	0.88	1.76	FUTURE	1	2.96	2.96	PRIOR	2	2.07	4.14	WERE	1	0.46	0.46
BOARD	1	3.38	3.38	GND	3	1.65	4.95	PUBLIC	3	5.64	16.92	WHO	1	2.29	2.29
BRU	1	8.22	8.22	HAD	3	0.50	1.50	QUARANTINED	1	9.83	9.83	WHOLE	1	4.18	4.18
BY	2	0.85	1.70	HE	2	0.99	1.98	RECOVERED	1	5.43	5.43	WKS	1	4.79	4.79
CALL	1	1.84	1.84	HEALTH	3	6.65	19.95	REFUSED	1	4.22	4.22	WOULD	1	1.10	1.10
CAPT	2	3.65	7.30	HRS	2	2.76	5.52	REMAIN	1	3.61	3.61	Y	1	2.99	2.99
CLRED	1	4.18	4.18	I	1	0.31	0.31	REQUESTED	1	2.30	2.30	Z	1	3.58	3.58
COMPANY	1	2.25	2.25	IN	4	0.27	1.08	ROW	1	4.69	4.69				
CONTACTED	1	2.46	2.46	KEEP	1	3.19	3.19	SAID	2	1.27	2.54				
CONTAGIOUS	2	9.14	18.28	KNOW	1	2.37	2.37	SAW	1	2.11	2.11				

Figure 3. ASRS Report ACN:642270 modeled as a bag of words

IV. Approach

In our approach, each document of the corpus is represented as a bag of words, thus discarding all document and sentence structure. For each document, the term frequency for each word is calculated. Then, for each term, a corpus-wide count of documents that contain the term is calculated. These two quantities are combined in a measure called TF*IDF, which stands for term frequency multiplied by inverse document frequency. TF*IDF is an intuitive weighting measure for the component terms in the document: the more often a term occurs in a document, the higher it is weighted; the more documents a term occurs in, the lower it is weighted. This matches our expectation that a term will be more significant when it occurs frequently in a document but in few documents of the corpus. There are a variety of ways to scale the term frequency and inverse document frequency components in TF*IDF; we use the most common formulation:

$$TF*IDF = term\ frequency * log(inverse\ document\ frequency) \quad (1)$$

When calculating the inverse document frequency, we used only the documents used to train the classifier (i.e., the training set) rather than the entire corpus, as we did for all our statistics used in training. As a result, the inverse

document frequency score would vary minimally from one run to another. Figure 3 gives an example of a document modeled as a bag of words.

We subdivided the corpus into separate training and testing sets. Ninety percent of the corpus (approximately eighteen thousand documents) was available for training and the remaining ten percent (approximately two thousand documents) were used for testing. We created a separate classifier for each class (so for a single run and method, forty separate classifiers were created). For the SVM experiments, only a fraction of the available testing data was used, for two reasons. First, training a SVM is relatively time consuming, and training over all the available training data was not feasible given the time constraints of our study. Second, it is often best to train a SVM classifier using equal proportions of positive and negative examples[19]. Therefore, we manipulated the training set so that equal numbers of positive and negative examples were present in the training data used. For many of the incident classes, this meant that we used a lower number of training examples when training the SVM- in some cases, fewer than four hundred documents. The ordering and selection of training data from the training set was randomized in all cases. The testing set was not altered in any way, so it preserved the overall distribution of positive to negative examples, with natural variations due to the randomness of the partitioning of the corpus.

We evaluated the results of our experiments using the standard measures of precision and recall rather than the rate of correctly classified instances. The rate of correctly classified instances can be misleading when there are few positive instances to classify- in such cases, a classifier can classify every instance as negative (thus misclassifying all positive instances) yet still have a high rate of correctly classified instances. Instead, precision is a measure of purity for the positively classified instances, defined as the ratio of the correctly classified positive instances to all positive instances in the testing set, and recall is a measure of coverage, defined as the ratio of the correctly classified positive instances to all instances classified as positive from the testing set. Consider the four types of classification outcomes: true positives (correctly classified positive instances), false positives (negative instances incorrectly classified as positive), false negatives (positive instances incorrectly classified as negative), and true negatives (correctly classified negative instances). Representing these as *tp*, *fp*, *fn*, and *tn* respectively, the formula for precision *p* is:

$$p = tp / (tp + fp) \tag{2}$$

and the formula for recall *r* is:

$$r = tp / (tp + fn) \tag{3}$$

Precision and recall are often combined into a third measure, F-measure[20] (also called F1-measure), defined as:

$$F = (2 * p * r) / (p + r) \tag{4}$$

which we use in our evaluation as well.

The measures of precision and recall (and correspondingly, F-measure) are highly sensitive to the number of true positives. With many of our target incident classes having few positive examples (see Figure 1), we could have twenty or fewer positive instances in our testing set. This presented the possibility of random effects skewing our results. We used three runs of ten-fold cross-validation to address this issue. Cross-validation amounts to rerunning the experiment several times with different training and testing data selected from the corpus. The corpus is divided into folds of approximately the same size. The experiment is run once for each fold, every time using a different fold as the testing data and using the remaining folds for training data. This insures that all data are used for testing, thus minimizing the potential for random effects. The results of the multiple runs are averaged together.

V. Experimental Methodology

We devised a battery of experiments to approximate the particular semantic transformations as well as several non-semantic methods for the purpose of comparison.

1) Baseline methods

We created three baseline methods against which to compare the results of our experiments, as well as to evaluate the performance of the SVM text classification approach itself. The first method, *Baseline*, was our generic SVM-based approach. We used the unaltered bag of words representation (as described in the previous section). We had explored using mutual information for feature selection (e.g., to select the terms), but found no significant

improvement over simply selecting the most frequently occurring terms in the corpus. This corresponds with prior research, which also found term selection provided no improvement in classification performance when using an SVM for text classification[21]. Favoring simplicity, we selected all terms that occurred in more than one percent of the documents of the training set (approximately 1600 terms) and discarded all less common terms, as they occurred in less than twenty documents in our training set and thus were unlikely to significantly contribute to classification accuracy. Stop words (i.e., words with little semantic value) were included in the list of terms. We used the Weka software package[22], version 3.4.7, for the SVM implementation, using an RBF kernel with a full cache and normalization. We used a reasonable starting point for the SVM parameters, based on ongoing experimentation in the domain[19], and did no further tuning. The *Baseline* method served as the basis for all of our SVM-based experiments, using the same SVM implementation and parameters, and the same bag of words with modifications as noted.

Our other two baselines did not use an SVM classifier and are the only methods we used that did not. The second method, *Feeble*, was meant to measure a performance floor for classification. Our *Feeble* classifier does not make use of training data, and as such, does not learn. Instead, it classifies all instances to be in the target class. Necessarily, the *Feeble* classifier will always have perfect recall, since all positive instances will be correctly classified. On the other hand, precision will match the ratio of positive instances to all documents in the training set. For most of our safety incident classes, this ratio is quite low, and the *Feeble* method performs poorly. Ideally, a reasonable classification technique will significantly outperform the *Feeble* method, but may not when the target class is difficult to characterize from the training data.

Our third method and final baseline, *BestWord*, used a simple scheme for classification. For each class, a single term is chosen as the lone classifying feature. When the chosen word is present in a document, it is classified as a positive instance of the class, and when the chosen word is absent, the document is classified as a negative instance. The magnitude of the frequency of classifying term in the document is not considered. For every class, each term was evaluated separately as the potential classifying word. The terms that had the best performance on the training set were selected as the classifying term. Since creating classifiers for the *BestWord* algorithm was much faster than training SVM classifiers, we were able to make use of all the available training data (approximately 18,000 instances). Unlike the SVM classifiers, the training sets were not manipulated to maintain a 1:1 ratio of positive to negative instances, as that was not likely to benefit the resulting *BestWord* classifier.

2) Pseudo-semantic methods

We created several methods to evaluate how different semantically-based transformations of the bag of words might impact classification performance, while using the same SVM parameters as *Baseline*. Such transformations require encodings of appropriate domain knowledge to guide the transformation. Unfortunately, domain specific resources were not available for our study, and general purpose semantic resources (such as WordNet[23], FrameNet[24] and the Open Mind Common Sense project[25]) are unlikely to include the specialized vocabulary and knowledge appropriate for aviation safety reports. Creating our own semantic resource was outside the scope of our study, so we approximated the sorts of transformations that an appropriate semantic resource could facilitate, as described below. As approximations, our methods could not firmly establish the benefit of a particular transformation with a proper semantic resource, but could guide us to approaches that held the most promise.

Our fourth method, *WordSplit*, was designed to measure the potential for exploiting synonymy within the documents, specifically by replacing all terms in a set of synonyms with a single term. For example, rather than using both “aircraft” and “airplane” in the text, only “aircraft” would be used. Our assumption was that restricting usage to a single term (rather than the full set of synonymous variations) would increase the number of uses of the term in the training corpus, thereby enhancing the classifier’s ability to generalize. Without a list of appropriate synonyms, it is difficult to see how merging synonyms terms can be simulated, so we simulated the reverse transformation. Every term in the corpus was randomly split into two artificial synonymous terms, with an equal probability for each synonymous form. For example, “aircraft” was split into “aircraft1” and “aircraft2”, so each occurrence of “aircraft” in the text was substituted with either “aircraft1” or “aircraft2”. Real synonyms may have different patterns of usage, but our artificial synonyms came from the same term and so had a stronger sense of equivalence than actual synonyms. TF*IDF computations and term selection was performed on this transformed corpus in the same manner as described previously. *WordSplit* was intended to reverse the merging of synonyms, so our expectation was that it would underperform *Baseline* if merging synonymous terms in the bag of words would be generally beneficial.

Our fifth method, *Collocation*, identified all bigrams (i.e., pairs of words occurring consecutively in the narrative) and added them as new terms into the bag of words. Thus, a phrase such as “hold short” would be represented as “HoldShort” in the bag of words, along with the component words “hold” and “short”. No attempt

was made to identify meaningful phrases or indicative bigrams, so many of the bigrams comprised of common words that occurred together but had no special significance. Some of the included bigrams have a special meaning in the domain (e.g., "hold short" means to stop an aircraft before some point). Including bigrams significantly increases the size of the bag of words representation: approximately twenty-four thousand bigrams occurred in more than one percent of the corpus, resulting in over four thousand features (terms and bigrams) selected. We expected that the benefits of including significant bigrams would offset the drawbacks of including insignificant bigrams, resulting in an increase in classification performance if phrase identification was particularly useful in this domain.

Our sixth method, *PartOfSpeech*, was designed to measure the effect ambiguous terms might have on classification. Disambiguating terms manually went beyond the scope of our study, and disambiguation is difficult to simulate or approximate. Similar to our *WordSplit* experiment, artificial ambiguation has been used previously to study the effects of disambiguation in an information retrieval context[26]. However, subsequent studies have contradicted the findings based on this technique, and critiqued artificial ambiguation in general as unrepresentative of ambiguity as it normally occurs in actual language[27,28]. For these reasons we avoided using artificial ambiguity and instead used the General Architecture for Text Engineering (GATE) package of tools[29] to perform a simple disambiguation based on parts of speech. This disambiguation is neither completely accurate nor as powerful as we would like, since only the distinction between nouns, verbs, adjectives and adverbs is provided and not finer the distinctions between word senses. Nonetheless, some important differences are identified. For instance, the noun "pilot" is represented separately from the verb "pilot" in the transformed bag of words, and such distinctions could be important for certain classes.

3) Term grouping methods

Our last set of methods was intended to estimate how the semantic relationship between terms might be exploited to improve classification performance. Terms, or more specifically, the concepts they represent, have a wide variety of potential relationships. For instance, "pilot" is a type of "person" who uses an "aircraft" to perform "flying." Domain knowledge such as this has been the mainstay of artificially intelligent applications, with ontologies recently emerging as a preferred representation. It may not be obvious how such a varied set of relationships could be factored into the text classifier, but one possibility would be to use the ontology as the basis for generalization. Specifically, the ontology could be used to generalize more specific terms into a broader term if there is no practical difference between the specific terms for purposes of classification. For example, a "doctor" and a "nurse" can both be represented as a "medical practitioner", and the difference between the two is not likely to be meaningful when classifying reports as medical emergencies. For another type of incident, however, the distinction may be relevant, so performing such transformations are not necessarily beneficial for all classes.

term	tf	idf	tf*idf	term	tf	idf	tf*idf	term	tf	idf	tf*idf	term	tf	idf	tf*idf
04	2	5.86	11.72	CONTAGIOUS	2	9.14	18.28	KEEP	1	3.19	3.19	ROW	1	4.69	4.69
1/2	2	4.93	9.86	COULD	1	1.33	1.33	KNOW	1	2.37	2.37	SAID	2	1.27	2.54
2	2	1.04	2.08	CREW	1	1.92	1.92	LEONE	1	9.83	9.83	SAW	1	2.11	2.11
3	1	1.53	1.53	CUSTOMS	1	5.33	5.33	LNDG	2	1.25	2.50	SIERRA	1	6.74	6.74
ADVISED	3	1.86	5.58	DEC	2	4.70	9.40	MALARIA	1	9.83	9.83	SO	1	1.40	1.40
AGENT	1	3.81	3.81	DISPATCH	1	2.97	2.97	MAY	2	2.37	4.74	STAFF	3	6.00	18.00
ALERTED	2	4.43	8.86	**DOCTOR**	2	5.12	10.24	ME	1	1.24	1.24	THAT	2	0.39	0.78
ALL	1	1.34	1.34	**pseudo1-10**	**2**	**3.75**	**7.50**	MEET	2	4.02	8.04	THE	1	0.08	0.08
ALSO	1	1.51	1.51	ENTIRE	1	3.50	3.50	**MET**	1	3.61	3.61	THOUGHT	1	2.14	2.14
AND	5	0.06	0.30	FLT	4	5.06	20.24	**pseudo31-70**	**1**	**1.05**	**1.05**	TO	4	0.03	0.12
ARRIVED	1	3.25	3.25	FLY	1	2.38	2.38	MGRS	1	7.19	7.19	TUE	1	4.25	4.25
ATTACK	1	5.48	5.48	FOR	1	0.37	0.37	MON	1	4.55	4.55	UNTIL	2	1.94	3.88
pseudo11-30	**1**	**2.40**	**2.40**	FRA	1	7.89	7.89	NOT	1	0.45	0.45	US	1	0.98	0.98
BE	2	0.88	1.76	FULLY	1	3.96	3.96	ON	2	0.20	0.40	WASN'T	2	0.45	0.90
BOARD	1	3.38	3.38	FUTURE	1	2.96	2.96	PAX	3	5.24	15.72	WE	2	0.46	0.92
BRU	1	8.22	8.22	GND	3	1.65	4.95	PRIOR	2	2.07	4.14	WERE	1	0.46	0.46
BY	2	0.85	1.70	HAD	3	0.50	1.50	PUBLIC	3	5.64	16.92	WHO	1	2.29	2.29
CALL	1	1.84	1.84	HE	2	0.99	1.98	QUARANTINED	1	9.83	9.83	WHOLE	1	4.18	4.18
CAPT	2	3.65	7.30	HEALTH	3	6.65	19.95	RECOVERED	1	5.43	5.43	WKS	1	4.79	4.79
CLRED	1	4.18	4.18	HRS	2	2.76	5.52	REFUSED	1	4.22	4.22	WOULD	1	1.10	1.10
COMPANY	1	2.25	2.25	I	1	0.31	0.31	REMAIN	1	3.61	3.61	Y	1	2.99	2.99
CONTACTED	1	2.46	2.46	IN	4	0.27	1.08	REQUESTED	1	2.30	2.30	Z	1	3.58	3.58

Figure 4. ASRS Report ACN:642270 with pseudowords added below the component word.

Our seventh method, *Pseudoword,* was intended to simulate what the effects of combining sets of related terms together would have on classification performance, such as representing both "doctor" and "nurse" as a "medical

practitioner" in the example above. For every set of related terms, an artificial pseudoword was created to represent all the terms in the corresponding set. The bag of words was modified to include the pseudoword whenever a component term was present. The occurring component terms were also left as is in the bag of words. Figure 4 shows an example of this transformation. The pseudowords were treated as normal terms for the purposes of the TF*IDF scaling computations. Since each pseudoword represented a set of terms, they occurred more frequently in the modified corpus, resulting in higher term frequencies and typically lower inverse document frequencies. This meant we could use less frequent component terms (we used component terms that occurred in as few as 0.1% of all documents in the corpus) for our pseudowords and still have enough occurrences of the pseudoword to make it viable for training.

As we did not have an appropriate ontology to use, we once again tried to simulate the desired transformation by approximating reasonable sets of terms to combine. Since we had no knowledge structure or relationship between terms to use, nor an easy way to explore the space of combining terms, we once again used an approximation. In this case, we used a measure of each term's correlation with the target class. Our supposition was that highly correlated terms would have some semantic relationship to one another, as "doctor" and "nurse" would for evaluating medical emergencies in the example above. However, since terms were identified only by their correlation to the target class and not through their meaning, they were not guaranteed to have any specific semantic relationship to each other (like a common hypernym in the example above). Indeed, the set of indicative terms are likely to incorporate a variety of semantic relationships amongst each other; so along with "doctor" and "nurse", one might also expect to see terms like "breathing" or "unconscious" in the list of terms associated with medical emergencies. A separate list of terms that were inversely correlated with the target class was also generated. For example, terms like "engine" or "restricted" might indicate that the incident was not a medical emergency. The semantic relationship between the terms that negatively correlate with the target class is even more tenuous, often including disparate concepts that are positively correlated to other classes.

pseudoword	word	f*prec	pseudoword	word	f*prec	pseudoword	word	f*prec
pseudo1-10	paramedics	0.29116		medication	0.04438		attendant	0.012
	doctor	0.29057		attack	0.04307		volunteered	0.01071
	paged	0.2287		sick	0.04107		lady	0.01033
	medical	0.21594		husband	0.02971		aisle	0.00988
	breathing	0.21164		ambulance	0.02588		laid	0.00984
	heart	0.20679		assisted	0.02444		onboard	0.00947
	physician	0.19911	pseudo31-70	patient	0.02398		passenger	0.00934
	unconscious	0.19383		assistance	0.02243		assist	0.00908
	AED	0.19322		enhanced	0.02066		her	0.00904
	nurse	0.18064		died	0.01823		diverted	0.00899
pseudo11-30	consciousness	0.1729		breath	0.01779		history	0.00891
	seizure	0.15183		symptoms	0.01687		traveling	0.00889
	administered	0.15007		complaining	0.01683		ASAP	0.00872
	pulse	0.14612		woman	0.01547		attended	0.00851
	ILL	0.135		CPR	0.01536		treatment	0.00821
	hospital	0.12777		shock	0.01504		contents	0.0078
	chest	0.11586		divert	0.01474		she	0.00761
	pains	0.10591		retrieved	0.01455		assisting	0.00754
	doctors	0.10241		conscious	0.01396		BOY	0.00752
	emt	0.10206		female	0.01354		family	0.00746
	blood	0.10147		mother	0.01339		row	0.00721
	kit	0.09692		MET	0.01337		diversion	0.00698
	oxygen	0.05819		pax	0.01337			
	pain	0.04812		attending	0.01243			

Figure 5. Creating three pseudowords from the ranking of component words.

We wanted to guide the pseudoword generation towards pseudowords that were highly correlated with the target class. Since each pseudoword was a conglomeration of its component words, we felt that selecting higher precision words, specifically by weighting precision higher than recall, would ultimately lead to more effective pseudowords. We used a measured similar to F0.5-measure (itself a variant of F-measure), which we call $F_{p,}$

$$F_p = F * p = (2 * p^2 * r) / (p + r) \quad (5)$$

though other measures could also be used. We chose to make six pseudowords for each target class: three from positively correlated terms and three from negatively correlated terms. The first pseudoword was created from the

ten most highly ranked terms; the second from the next twenty ranked terms; and the third from the next forty ranked terms. Figure 5 gives an example of how these pseudowords were formed.

Our eighth and final method, *NoGroup*, served as a comparison to *Pseudoword*. We wanted to know whether any observed difference in performance between *Pseudoword* and *Baseline* were due to creation of the pseudowords or the way the pseudowords were selected, specifically by the inclusion of lower frequency terms selected by our measure F_p. Therefore, *NoGroup* used the same unaltered bag of words as *Baseline*, additionally including any rare component words from *Pseudoword* that were omitted in *Baseline*.

VI. Empirical Results

The results of our experiments are given in Figure 6, Figure 7, and Figure 8. The incident classes are ordered as in Figure 1, in order of decreasing prevalence from left to right. Regardless of the method used, there is a general tendency for decreased performance as the incident classes decrease in prevalence. This is an expected result, as precision will usually decline when fewer positive instances are in the testing set; forcing a 1:1 positive to negative ratio on the testing set would eliminate the downward trend. Likewise, recall, which is not affected by the scarcity of positive instances, is more or less stable across the incident classes. Furthermore, the SVM methods had less training data available for rare incidents, as we had forced a 1:1 ratio of positive and negative instances in the training set.

Figure 6 shows the results of our three baselines in terms of precision, recall, and F-measure. The average measures over all incident classes for *Baseline*, *Feeble*, and *BestWord*, respectively, are: precision, 0.24, 0.08, and 0.33; recall, 0.81 1.00, and 0.50; and F-measure, 0.34, 0.13, and 0.36. As expected, the *Baseline* method significantly outperforms the *Feeble* method. However, in some cases the performance was similar and for a single incident class, *Feeble* actually outperformed *Baseline*. The classes where *Feeble* compared favorably with *Baseline* tended to be "catch-all" classes (i.e., containing all reports that did not fit a more specific class) or very broad classes that are difficult to characterize. It is not surprising that a classifier would have very poor performance on such classes. What is surprising is the competitive performance of *BestWord* when compared to our *Baseline*. *BestWord*'s overall performance actually exceeded *Baseline*, typically underperforming on the frequent incident classes but outperforming *Baseline* on the less frequent incident classes. The performance of the *BestWord* method does not appear to be as tightly coupled with the frequency of the positive instances, which gives it an overall advantage over *Baseline*. Although tuning of the SVM parameters and careful selection of terms could boost the performance of *Baseline*, the relative success of the *BestWord* method shows that relatively simple methods can perform competitively, particularly when overall performance is low.

Figure 7 shows the results of our pseudo-semantic methods along with the comparative *Baseline* in terms of precision, recall, and F-measure. The average measures over all incident classes for *WordSplit*, *Collocation*, and *PartOfSpeech*, respectively, are: precision, 0.22, 0.23, and 0.24; recall, 0.79, 0.77, and 0.81; and F-measure, 0.32, 0.33, and 0.34. The three methods tracked the *Baseline* method closely and exhibited only insignificant differences in performance. For *WordSplit*, we had artificially split each word into two pseudo-synonyms and had expected that performance would significantly decrease if merging synonyms are beneficial. *WordSplit* did show slightly lower performance, but not enough for us to conclude that it would be worthwhile to combine synonyms into a single term. *Collocation* also showed slightly worse performance than the *Baseline*. We had expected that the inclusion of meaningful phrases such as "hold short" would boost performance, but performance actually decreased when we included bigrams – possibly because most of our bigrams were not actual phrases but only terms that frequently occur consecutively in a sentence. Regardless, the results of the *Collocation* experiment fail to indicate any advantage to the inclusion of phrases. Finally, the *PartOfSpeech* method did outperform *Baseline* but the difference between the two is not significant. The bag of words representation makes it difficult, if not impossible, to fully reconstruct the meaning of the original document. Rather, it provides an overall gist of the document, and in that context, the difference between noun and verb forms (as in "pilot") may not be relevant in this domain. In any case, the results of the *PartOfSpeech* experiment fail to show that disambiguation would be worth the effort.

Figure 8 shows the results of our term grouping methods along with the comparative *Baseline* in terms of precision, recall, and F-measure. The average measures over all incident classes for *Pseudoword*, and *NoGroup*, respectively, are: precision, 0.27, and 0.24; recall, 0.79, and 0.81; and F-measure, 0.39, and 0.35. The *Pseudoword* method compares favorably with the *Baseline* method, offering equal or better performance, as measured by F-measure, in all cases. (*Baseline* had a slightly better F-measure value for a few incident classes but the difference is not significant.) Moreover, *Pseudoword* had the most significant performance gains over *Baseline* for the rare incident classes. On the whole, the *Pseudoword* method produced higher precision, particularly for these rare incident classes, in exchange for a loss of recall. *NoGroup* also performed favorably when compared to *Baseline*. This indicates that the inclusion of the high precision, low frequency words is beneficial and presumably contributes

to the performance gains of *Pseudoword* over *Baseline*. However, *Pseudoword* significantly outperforms *NoGroup* as well. This leads us to conclude that combining indicative terms into pseudowords boosted classification performance in this domain.

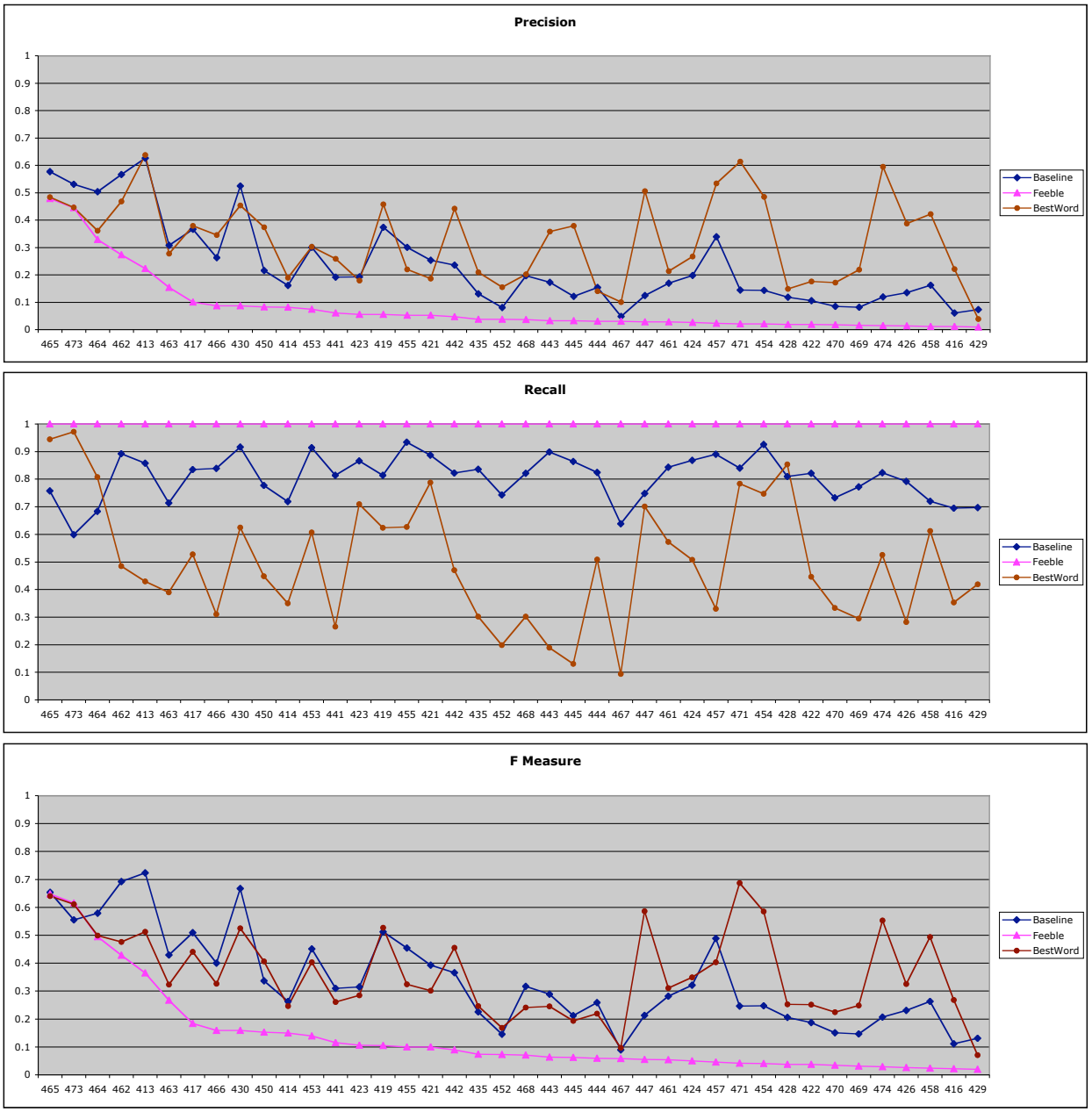

Figure 6. Results of the *Baseline*, *Feeble*, and *BestWord* experiments

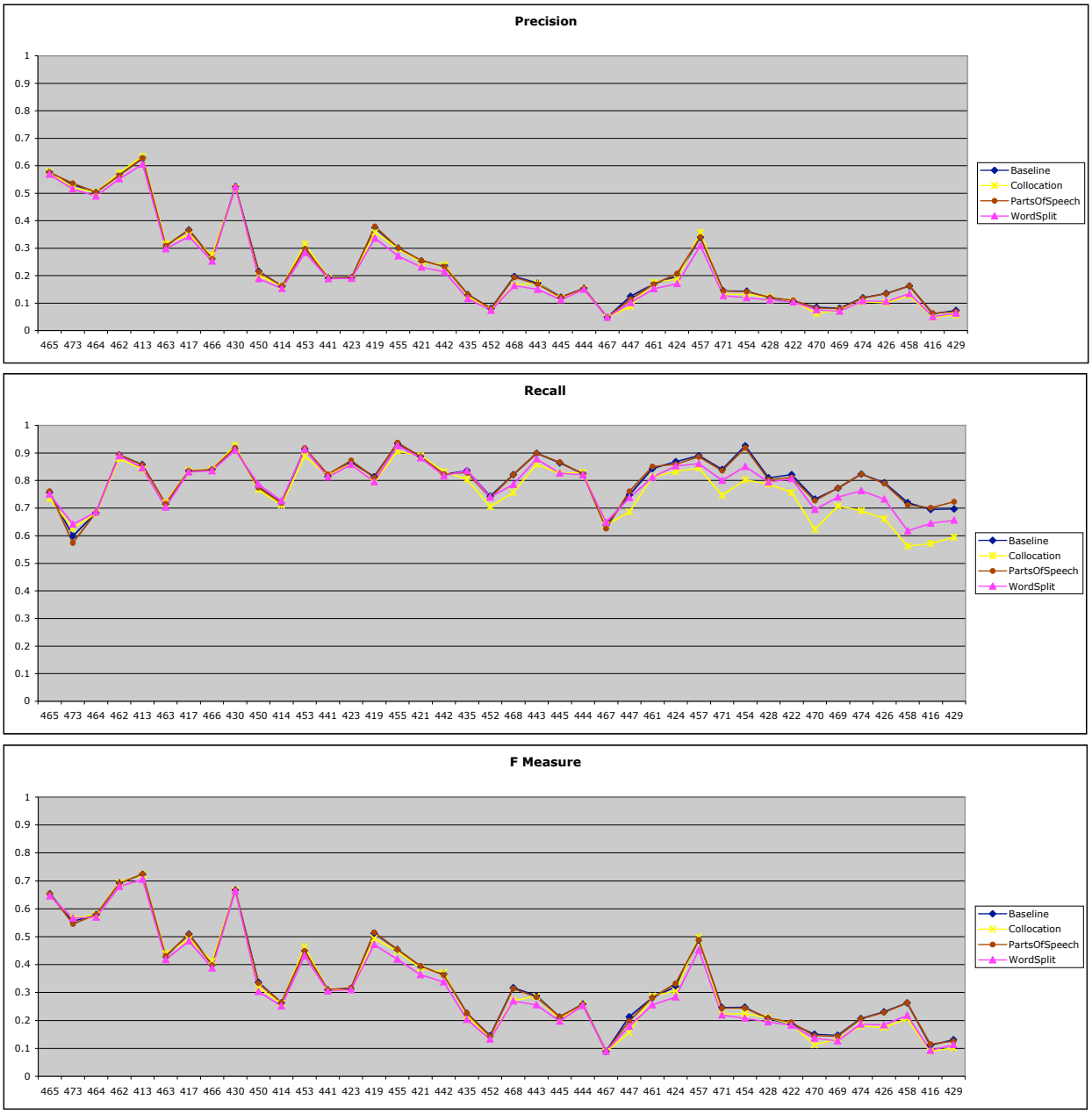

Figure 7. Results of the *WordSplit*, *Collocation*, and *PartOfSpeech* experiments.

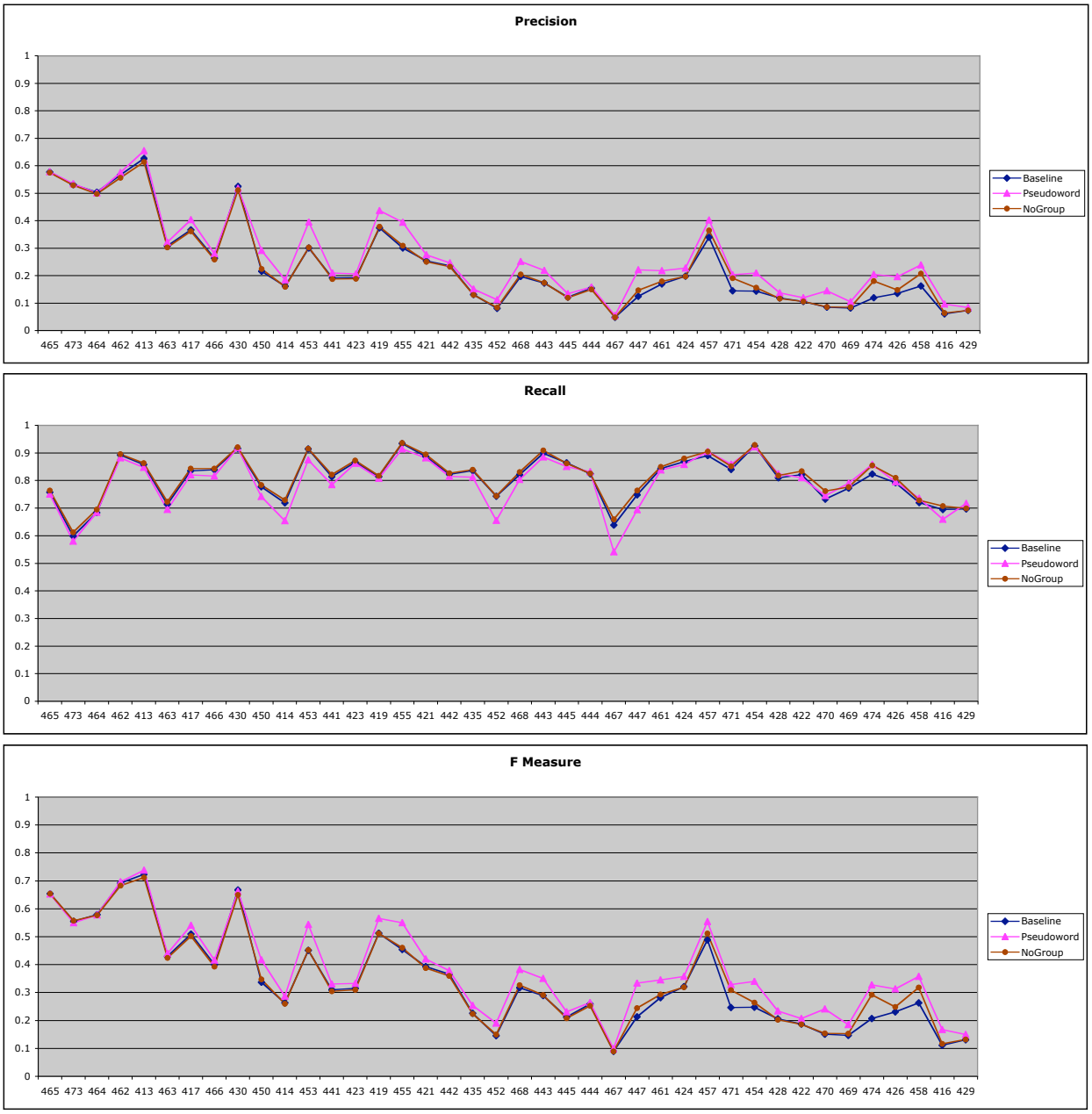

Figure 8. Results of the *Pseudoword* and *NoGroup* experiments.

VII. Conclusions and Future Work

Text classification today makes little explicit use of semantics. This would appear to be an obvious shortcoming, and we hypothesized that the incorporation of semantics into standard text classification methods could easily result in significant improvement in classification performance. Unfortunately, appropriate semantic resources were not available as a part of our study and are costly to create. Therefore, we developed a series of experiments that were meant to approximate what an appropriate semantic resource might contribute. For the most part, the results of these experiments do not support our hypothesis. In particular, merging terms based on synonymy, use of bigrams, and simple word disambiguation did not produce significant gains in classification performance. Our methods were approximations of semantic approaches and only applied to one domain, so we cannot reject our hypothesis in general. However, it does appear clear that introducing arbitrary semantic content into the bag of words model is not likely to produce positive results.

The results of our experiments were not entirely negative. We used a simple classification scheme, *BestWord*, which compares favorably with our SVM-based baseline. An ensemble learner that uses both *BestWord* and the more standard SVM-based approach could produce better overall performance. The *BestWord* method could also be expanded to use multiple terms in its classification. Our intuition is that ultimately more sophisticated methods should be able to outperform *BestWord* or any subsequent variants, but it serves as a reasonable point of comparison. Also, due to its simplicity, *BestWord* is easily understood and very fast, making it an easy platform to explore transformations that could lead to improvements in the bag of words model.

We used a term ranking measure, F_p, to identify high precision terms that could improve classification results. Even with limited usage, our use of this measure to select rare but valuable terms in the *NoGroup* method produced positive results. Further evaluation is needed to determine if F_p is useful in general, as a basis to select all the terms as well as in other text classification domains. F_p should also be compared to commonly used measures, such as mutual information.

The most promising result from our experiments was the performance gains of the *Pseudoword* method over our *Baseline* method. Many of the choices in the *Pseudoword* method were arbitrary and the grouping technique used was crude from a semantic perspective, but *Pseudoword* still significantly outperformed our baselines. This shows that grouping of indicative terms may be beneficial, but more evaluation and refinement of the *Pseudoword* method is warranted. First, the number and size of the term groupings should be examined. Second, the actual semantic relationships should be identified in the term groupings, and the component terms should be recombined into sets that adhere to these semantic relationships. Third, the *Pseudoword* method should be evaluated with an improved baseline with tuned SVM parameters and term selections. Finally, *Pseudoword* should be tried in other domains to see if the benefit is globally realized or particular to our corpus of ASRS reports.

Our initial hypothesis, that incorporating semantics into the bag of words would improve classification results, is difficult to disprove, as the possibilities for incorporating semantics are vast. In addition, we approximated what effects a truly semantic approach might have, and so it is quite possible that merging synonyms, phrasal identification and term disambiguation can improve classification results, even in our domain. However, the results of our study failed to show a benefit to these approaches, so discretion should be used before expending significant effort to develop the appropriate semantic resources. Increasing classification performance by infusing additional information into the bag of words has proved difficult in our domain, as is reasonable classification performance in general. However, the *Pseudoword* method shows that classification performance gains can be achieved through manipulation of the bag of words, and may lead to a beneficial method for incorporating semantics into a text classification framework.

Acknowledgements

The author would like to thank Dr. Richard Keller, Francis Enomoto, Robert Carvalho, Joseph Castle and Dr. Rodney Martin for their contributions and Drs. Irv Statler, Ashok Srivastava and Deepak Kulkarni for their support of this work.

References

[1]NASA, "Aviation Safety Reporting System (ASRS) Program Overview," 2007: http://asrs.arc.nasa.gov/overview_nf.htm
[2]Connell, L. J., "ASRS Update and ASAP-to-ASRS Secure Data Transmission Protocols," *Presentation to the Maintenance ASAP InfoShare Working Group*, 2006: http://parks.slu.edu/departments/avsc/MEASAP/Documents/STLInfoShare/ASRSNASA.pdf
[3]Federal Aviation Administration, "ASRS System Level Business Rules," 2007: http://www.asias.faa.gov/pls/portal/docs/PAGE/ASIAS_PAGES/BUSINESS_RULES/ASRS_BR.html
[4]Salton, G. and Buckley, C., *Term-weighting approaches in automatic text retrieval*, Information Processing and Management **24** (5), 513 (1988).
[5]Salton, G. and Leck, M. E., *Computer Evaluation of Indexing and Text Processing*, Journal of the ACM (JACM) **15** (1), 8 (1968).
[6]Joachims, T., "Text categorization with Support Vector Machines: Learning with many relevant features," *ECML-98, Tenth European Conference on Machine Learning*, 1998.
[7]Sebastiani, F., in *Text Mining and its Applications*, edited by Zanasi, Alessandro (WIT Press, Southampton, UK, 2005), pp. 109.
[8]Megaputer, "Application of PolyAnalyst to Flight Safety Data at Southwest Airlines," Project Report, January 2004.
[9]Srivastava, A. N., Akella, R., Diev, V. et al., "Enabling the Discovery of Recurring Anomalies in Aerospace Problem Reports using High-Dimensional Clustering Techniques," *IEEE Aerospace Conference*, IEEE, 2006.
[10]Castle, J. P., Stutz, J. C., and McIntosh, D. M., "Automatic Discovery of Anomalies Reported in Aerospace System: Health and Safety Documents," *AIAA Infotech@Aerospace*, 2007.
[11]Péladeau, N. and Stovall, C., "Application of Provalis Research Corp.'s Statistical Content Analysis Text Mining to Airline Safety Reports," Project Report, February 2005.
[12]SRA International, "Text Mining of Pilot Report Narratives at United Airlines," Project Report, November 27 2000.
[13]Posse, C., Matzke, B., Anderson, C. et al., "Extracting information from narratives: an application to aviation safety reports," *IEEE Aerospace Conference*, 2005.
[14]Maille, N., Statler, I. C., and Chaudron, L., "An Application of FCA to the Analysis of Aeronautical Incidents," *International Conference on Formal Concept Analysis*, 2005.
[15]Ananyan, S. and Goodfellow, M., "New Capabilities of PolyAnalyst Text and Data Mining Applied to STEADES Data at the International Air Transport Association (IATA)," Project Report, October 2005.
[16]Manning, C. D. and Schütze, H., *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, USA, 1999.
[17]Koller, D. and Sahami, M., "Hierarchically Classifying Documents Using Very Few Words," *Fourteenth International Conference on Machine Learning*, Morgan Kaufmann, 1997.
[18]NASA, "Cabin Attendant Reports," ASRS Database Reports, January 23 2007.
[19]Castle, J. P. (Personal Communication).
[20]Van Rijsbergen, C. J., *Information Retrieval (2nd edition)*, Second ed., Butterworths, London, 1979.
[21]Brank, J., Grobelnik, M., Milic-Frayling, N. et al., "Interaction of Feature Selection Methods and Linear Classification Models," *ICML-02 Workshop on Text Learning*, 2002.
[22]Witten, I. H. and Frank, E., *Data Mining: Practical machine learning tools and techniques*, 2nd ed., Morgan Kaufmann, San Francisco, 2005.
[23]Miller, G. A., *WordNet: A Lexical database for English*, Communications of the ACM **38**, 39 (1995).
[24]Baker, C. F., Fillmore, C. J., and Lowe, J. B., "The Berkeley FrameNet Project," *COLING-ACL*, 1998.
[25]Singh, P., Lin, T., Mueller, E. T. et al., "Open Mind Common Sense: Knowledge acquisition from the general public," *First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, Springer-Verlag, 2002.
[26]Sanderson, M., "Word sense disambiguation and information retrieval," *17th International Conference on Research and Development in Information Retrieval*, 1994.
[27]Gonzalo, J., Verdejo, F., Chugur, I. et al., "Indexing with WordNet synsets can improve text retrieval," *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, 1998.
[28]Hotho, A., Staab, S., and Stumme, G., "Wordnet improves text document clustering," *SIGIR Semantic Web Workshop*, 2003.
[29]Cunningham, H., Maynard, D., Bontcheva, K. et al., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," *40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.