Enhancing Sensitivity Classification with Semantic Features using Word Embeddings

Graham McDonald[1], Craig Macdonald[2], Iadh Ounis[2]

School of Computing Science
University of Glasgow, G12 8QQ, Glasgow, UK
[1]g.mcdonald.1@research.gla.ac.uk
[2]firstname.lastname@glasgow.ac.uk

Abstract. Government documents must be reviewed to identify any *sensitive* information they may contain, before they can be released to the public. However, traditional paper-based sensitivity review processes are not practical for reviewing born-digital documents. Therefore, there is a timely need for automatic sensitivity classification techniques, to assist the digital sensitivity review process. However, sensitivity is typically a product of the relations between combinations of terms, such as *who said what about whom*, therefore, automatic sensitivity classification is a difficult task. Vector representations of terms, such as word embeddings, have been shown to be effective at encoding latent term features that preserve semantic relations between terms, which can also be beneficial to sensitivity classification. In this work, we present a thorough evaluation of the effectiveness of semantic word embedding features, along with term and grammatical features, for sensitivity classification. On a test collection of government documents containing real sensitivities, we show that extending text classification with semantic features and additional term *n*-grams results in significant improvements in classification effectiveness, correctly classifying 9.99% more sensitive documents compared to the text classification baseline.

1 Introduction

Freedom of Information (FOI) laws[1,2] legislate that government documents should be opened to the public. However, many government documents contain *sensitive* information, such as *personal* or *confidential* information, that would be likely to cause harm to, or prejudice the interests of, an individual or organisation if the information were to be made public. Therefore, FOI laws provide exemptions that negate the obligation to release information that is of a sensitive nature.

To ensure that sensitive information is not made public, all government documents must be manually *sensitivity reviewed* prior to release. However, with the adoption of digital technologies, such as word processing and emails, the volume of government documents has increased and, moreover, documents are produced and stored in a more ad-hoc manner than the paper-based filing systems of previous decades. Therefore, the traditional sensitivity review process is not practical for the era of born-digital documents, and governments are facing an increasing backlog of digital documents awaiting review before they can be considered for release.

[1] http://www.legislation.gov.uk/ukpga/2000/36/contents [2] http://www.foia.gov

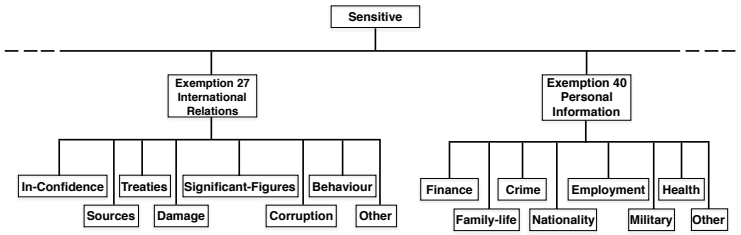

Fig. 1. The range of potential sensitivities relating to 2 of the 24 Freedom of Information Act 2000 (FOIA) exemptions, namely *International Relations* and *Personal Information*.

There is, therefore, a timely need for automatic sensitivity classification, to assist the digital sensitivity review process [1]. However, automatic sensitivity classification is a difficult task. For example, the UK Freedom of Information Act 2000 (FOIA) has 24 FOI exemptions[3], each with wide-ranging sub-categories of exemptions. Figure 1 illustrates the scope of potentially sensitive information from just 2 of these 24 exemptions, namely *International Relations* and *Personal Information*. As can be seen from Figure 1, the scope of potentially sensitive information is broad. Moreover, a document can, potentially, contain many unrelated sensitivities. Therefore, in this work, we view sensitive information as a *composite* class of information that can be a result of one or more different types of sub-category sensitivities.

Text classification [2] is one approach that has been shown to be promising as a basis for automatic sensitivity identification algorithms [3, 4]. Usually, a text classification model is learned by observing statistical patterns in the distributions of individual key terms from example documents. However, the potential effectiveness of sensitivity classification from single-term observations is limited, due to the fact that sensitivity classification is not a *topic-oriented* task [4] and, moreover, sensitivity tends to arise as a product of specific factors. For example, International Relations sensitivities are often a product of *who said what about whom*. It is, therefore, the relations between terms that can result in information being sensitive. One approach that has been shown to be effective at capturing the semantic relations between terms is word embeddings [5]. Word embeddings are vector space word representations, where each dimension maps to a latent feature of the word. We expect word embedding to be able to identify latent sensitivity in terms, due to two fundamental properties. Firstly, semantically *similar* terms are positioned close to each other within the vector space and, secondly, the directionality between multiple terms in the vector space can encode relations between the terms. Therefore, relations such as the previous example, *who said what about whom*, can have their relations preserved in specific dimensions of vector representations.

In this work, we present a thorough evaluation of the effectiveness of semantic word embedding features for sensitivity classification. On a test collection of government documents with real sensitivities, we compare semantic features with grammatical features derived from sequences of part-of-speech tags (POS) and term *n*-gram features. The contributions of this paper are two-fold. Firstly, we present the first in depth analysis of the effectiveness of word embeddings for sensitivity classification. Secondly, we show that semantic word embedding features can significantly improve the effectiveness of sensitivity classification. The combination of semantic word embeddings

[3] 14 of the 24 FOIA exemptions apply to documents that are to be archived for public access.

and term *n*-gram features correctly classified 9.99% more sensitive documents than the baseline text classification approach.

The remainder of this paper is structured as follows. In Section 2 we present work relating to sensitivity classification and word embeddings for text classification. In Section 3, we present the feature sets that we evaluate for sensitivity classification before, in Section 4, presenting our experimental setup. We present our results in Section 5, before providing some further analysis in Section 6, and conclusions in Section 7.

2 Related Work

Classifying sensitivities, such as FOI exemptions, to assist the sensitivity review of government documents, is a relatively new task. Moreover, it can be considered that the definition of sensitivity, in this context, is more broad than in most of the previous literature, e.g. preserving the privacy of personal data [6, 7]. McDonald *et al.* [3] was the first work to address the automatic classification of FOI exemptions. In that work, the authors presented a proof-of-concept classifier for classifying specific FOI exemptions, and found that extending text classification with additional features, such as *the number of subjective sentences* and a *country risk score*, could improve the effectiveness of text classification for specific sensitivities. The work that we present in this paper differs from the work of [3] in a number of ways. Firstly, in [3], the authors deployed individual classifiers for each specific sensitivity, whereas our work addresses the more challenging task of classification of the composite class of sensitivity. Secondly, in [3], the authors extended text classification with *hand-crafted* features that were tailored for specific sensitivities. In this work, we present a fully automatic approach that could easily generalise to other collections or sensitivities.

Berardi *et al.* [4] built on the work of McDonald *et al.* [3] to optimise the cost-effectiveness of sensitivity reviewers. In that work, Berardi *et al.* deployed a *utility-theoretic* ranking approach for semi-automatic text classification [8]. Their approach ranks documents by the expected gain in accuracy that a classification system can achieve by having a reviewer correct mis-classified instances, i.e. if a reviewer validates a document that the classifier is least confident about, then the overall accuracy is increased. Berardi *et al.* found that their approach performed well at estimating the correctness of classification predictions from McDonald *et al.*'s approach, and achieved substantial improvements in overall classification (+3% - +14% F_2). However, these improvements were much smaller than their approach had achieved on other tasks and they concluded that the task of classifying by sensitivity is much harder than *topic-oriented* classification.

In other work, relating to FOI exemptions, McDonald *et al.* [9] investigated methods for identifying passages of text in documents that contained information that had been supplied *in confidence*. In that work, the authors identified confidential information by measuring the amount of sensitivity in specific part-of-speech (POS) *n*-grams. Inspired by the work of Lioma and Ounis [10], who showed that high frequency POS *n*-grams have a greater *content load*, McDonald *et al.* used POS *n*-grams with a high *sensitivity load* to train a Conditional Random Fields sequence tagger for predicting confidential sequences. Their work showed that POS *n*-grams could be effective for identifying a specific sensitivity. Therefore, we also use POS *n*-grams as classification features in this work. However, differently from the work of McDonald *et al.* [9], we test if POS *n*-grams are effective features for classifying the *composite* class of sensitivity and compare POS *n*-grams with the performance of word embeddings and term features.

As previously stated in Section 1, word embeddings are vector space representations of terms [5]. Word embeddings have low dimensionality, compared to the sparse vector representations more traditionally used in text classification. The dense vector formation of word embedding models allow them to capture semantic qualities of, and relations between, terms in a collection. This has resulted in word embeddings becoming very popular in natural language processing tasks, e.g. [11, 12]. Moreover, there are a number of available word embedding frameworks, such as word2vec [13] and Glove [14], with models that are pre-trained on large corpora from different domains, such as Google News[4] or Wikipedia[5].

Recently, word embeddings have been shown to be effective in Information Retrieval and classification tasks, e.g. [15–17]. However, for classification, they have mostly been used for classifying short spans of text, such as tweets or sentences [17, 18]. Typically, word embeddings have been used as an initialisation step for neural networks. However, recently, Balikas and Amini [19] presented a large scale study that integrated word embeddings as classification features for multi-class text classification. In that study, the authors obtained document vector representations by deploying simple composition functions (e.g. min, average, max) to construct vector representations of combinations of words, such as phrases or sentences, from term vector models [20]. They showed that these compositional document vectors could be effectively used as features to extend text classification and improve classification performance. In this work, we follow the methodology of [19, 20] and compose document representations from word embeddings in the task of sensitivity classification. However, differently from Balikas and Amini [19], we show how these document representations combined with text features can be effective for discovering latent sensitivities.

3 Sensitivity Classification

In this section, we provide an overview of the feature sets that we test for sensitivity classification. Firstly, since term n-grams have not previously been studied for sensitivity classification, in Section 3.1, we briefly describe extending text classification with term n-gram features before, in Section 3.2, presenting the approach we deploy for generating grammatical features from POS sequences. Lastly, in Section 3.3, we present the approach that we deploy for generating semantic features using word embeddings.

The expected volumes of individual types of sensitivity vary between specific government departments. For example, in the UK, the Foreign and Commonwealth Office encounters many more *International Relations* sensitivities than the Department of Health. The approaches that we present in this section only depend on the terms in a collection and require no prior knowledge of specific sensitivities. Therefore, they could be deployed as part of a *first line of defense* across government departments.

3.1 Term Features

The first set of features that we evaluate are term features. Term features are a popular type of feature used for classifying textual documents. Indeed, using the frequencies of terms in documents to train classifiers, such as Support Vector Machines (SVM) [21], can be effective for many topic-oriented classification tasks [2].

[4] https://code.google.com/archive/p/word2vec/ [5] http://nlp.stanford.edu/projects/glove/

Although sensitivity classification is not a topic-oriented task [4], text classification has been shown to be a strong baseline approach [3, 4]. A popular, and effective, extension to text classification is to include additional *n*-gram term features [2]. *N*-gram features for text classification are, typically, a tuple of *n* contiguous terms from a larger ordered sequence of terms. Typically, text classification is extended with *n*-grams where $n \leq 4$. However, for sensitivity classification, we expect larger values of *n* to be more effective, since they have the potential to capture document structures that, in turn, can be an indicator of potential sensitivity. For example, table headings, such as *Name, Date of Birth, Residence*, can be a reliable indicator of Personal Information sensitivity. Therefore, in this work we test the effectiveness of larger term *n*-gram sequences, along with additional combinations of smaller values of *n* for completeness.

3.2 Grammatical Features

As previously mentioned in Section 2, part-of-speech (POS) *n*-grams have been shown to be effective for identifying text relating to *information supplied in confidence* [9]. However, as outlined in Section 1, sensitivity is a composite class containing many, more specific, types of sensitive information (such as confidential information) and the effectiveness of POS *n*-grams as features of sensitivity has not been fully studied for sensitivity classification. Therefore, in this work, we evaluate the effectiveness of POS *n*-grams as grammatical features for sensitivity classification.

POS *n*-gram features are derived similarly to the approach for term *n*-gram features. However, prior to selecting *n*-grams, a document is represented by the POS tags it contains. For example, the sentence "The informant provided the information" can be represented by the following POS tags "DT NN VB DT NN". When represented as POS 2-grams, the sentence becomes "DTNN NNVB VBDT DTNN". POS tags substantially reduce the vocabulary of a collection and provide a single representation of similar sentences. For example, sentences that are *about* different entities and actions but have the same grammatical structure have a single representation.

3.3 Semantic Features

In this section, we present the approach that we deploy for extending text classification with semantic features using word embeddings. As previously mentioned in Section 1, sensitivity is often a product of a combination of factors, such as *who said what about whom*. The common factors of these types of sensitivity are two-fold: Firstly, relations between terms are often preserved over multiple sensitivities. For example, in the sentences "the assailant denied offering the plans for the attack" and "The informant provided us the names of the suspect" the relation of Entity A giving something to Entity B is common to both sentences; The second common factor is that the entities or actions often have similar meaning, e.g. offering/provided or informant/assailant.

Word embedding models are trained by observing the contexts in which terms usually appear within large corpora, with the assumption that words occurring within similar contexts are semantically similar. The resulting word embedding models have two fundamental properties that can help us to identify relational sensitivities. Firstly, semantically similar terms tend to appear close to each other in the vector space (e.g. informant/assailant) and, secondly, the directionality between terms in the vector space can

Table 1. Experimental Setup: Feature set combinations and abbreviations.

Feature Set	Stand Alone	Extending Baseline
Text Classification (baseline)	Text	-
Term *n*-grams	TN	Text+TN
Grammatical	POS	Text+POS
Semantic	WE	Text+WE
Term & Grammatical	TN+POS	Text+TN+POS
Term & Semantic	TN+WE	Text+TN+WE
Grammatical & Semantic	POS+WE	Text+POS+WE
Term & Grammatical & Semantic	TN+POS+WE	Text+TN+POS+WE

encode relations between terms (e.g. the direction of *assailant* to *offering* is close to parallel with *informant* to *provided*). This, in turn, means that semantically similar relations tend to have similar values in specific dimensions of their embedding representations.

To derive semantic features, we follow the approach of Balikas and Amini [19] to construct a document representation from word embeddings using a set of composition functions, *min*, *mean* and *max* [22, 23]. For a given word embedding model, W, of term vectors, $V^{\text{term}} \in W$ and a document collection, C, a document vector representation, $V^{\text{doc}}, |v^{\text{doc}}| = |v^{\text{term}}|$, is composed by applying a composition function, $F \in \{min, mean, max\}$ to each document, $d \in C$. For example, using the composition function F_{max}, the value of the *n*th dimension of the document representation, denoted as $V^{doc}_{d,n}$, is:

$$V^{doc}_{d,n} = max(V^{term}_{i,n}) \forall i \in C_d \tag{1}$$

Each dimension of V^{doc} can then be used as a single feature for the purposes of classification. Moreover, in addition to the composition functions *min*, *mean* and *max*, we also deploy the compound function *concat*, where the resulting document representation is:

$$Concat(d) = [min(d), mean(d), max(d)] \tag{2}$$

Word embedding models capture the semantic relations of terms *within a collection*. Therefore, it is possible that semantic relations which are important for identifying sensitivities within our test collection may not be present in our chosen model. To address this, we construct document representations using two word embedding models that have been trained on different domains, namely Google News[6] and Wikipedia[7]. To do this, we apply the selected composition function, F, to each model, w_i, separately, to obtain a document representation from each model. We concatenate the document representations and use each vector dimension as a separate classification feature, resulting in the document representation:

$$semantic\_representation(d) = [F(w_i, d), F(w_{(i+1)}, d), ...F(w_n, d)] \tag{3}$$

4 Experimental Setup

In this section we present our experimental setup for evaluating the effectiveness of *term*, *grammatical* and *semantic* features for sensitivity classification. The research questions that we address are two-fold. Firstly, **RQ1**: "Are semantic word emdeddings features more effective for sensitivity classification than grammatical or term features?" and, secondly, **RQ2**: "Does using multiple word embedding models trained on different

[6] https://code.google.com/archive/p/word2vec/ [7] http://nlp.stanford.edu/projects/glove/

domains further improve the effectiveness of semantic features for sensitivity classification?". Table 1 presents the combinations of feature sets that we evaluate, and the abbreviations that we use to denote each combination in the remainder of this paper.

Collection: We use a test collection of 3801 government documents that contain real sensitivities. The documents were sensitivity reviewed by trained government sensitivity reviewers, who assessed the documents against 2 FOIA exemptions, namely *International Relations* and *Personal Information*. All documents that were judged as containing any Exemption 27 or Exemption 40 sensitivities were labeled as *sensitive*. Table 2 presents the resulting collection statistics, after stopword removal. We use a 5-fold Cross Validation to perform the binary classification *sensitive* vs. *not-sensitive*. To address the class imbalance in the collection (13.2% sensitive), we match the number of sensitive and not-sensitive training instances by randomly down-sampling the *not-sensitive* documents in each fold.

Table 2. Salient statistics of our test collection.

Total Documents	**Not Sensitive**	**Sensitive**				**Unique Terms**	**Avg. Doc Length**
		International Relations	Personal Information	Both	Total		
3801	3299	231	156	115	502	122 348	710 terms

Baseline: We evaluate each of the feature sets against a baseline text classification system using bag-of-words uni-gram term features, denoted as Text. We remove stopwords and terms that appear in only 1, or more than half, of the training documents in a fold. Feature values are binary, i.e. term features are either present or not. When extending text classification, additional features are scaled in the range $[0, 1]$.

Term Features: For term features, presented in Section 3.1, we test for term *n*-grams where $n = \{2..10\}$. When testing for values of *n*, we include *n*-grams for all values $< n$, i.e. when $n = 3$ feature vectors are constructed from all bi-grams and tri-grams. In the remainder of this paper, we denote term features as TN_n (i.e. for the previous example, TN_3). Feature values are binary, i.e. either present or not.

Grammatical Features: For grammatical features, presented in Section 3.2, we use the TreeTagger[8] part-of-speech tagger to POS tag documents and use a reduced set of 15 POS tags following [9, 10]. We test for POS *n*-grams where $n = \{1..10\}$. Following the experimental setup for term features, when testing for values of *n*, we include *n*-grams for all values $< n$. Grammatical features are denoted as POS_n.

Semantic Features: We use *pre-trained* word embedding models and test if using two word embeddings models trained on different domains improves the effectiveness of semantic features for sensitivity classification.

Table 3. Pre-trained word embedding models for deriving semantic features.

Model	**Architecture**	**Vocabulary Size**	**#Dimensions**	**Training**	**Context Window**	**Ref**
Google News	word2vec	3M	300	Negative Sampling	$\text{BoW}\tilde{5}$	WE_gn
Wikipedia+Gigaword5	Glove	400,000	300	AdaGrad	10+10	WE_wp

[8] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

Table 4. Results for combinations of *textual*, *grammatical* and *semantic* feature sets, compared against the text classification baseline.

Configuration		precision	TPR	TNR	F_1	F_2	BAC	auROC
Text		0.2410	0.6573	0.6841	0.3520	0.4874	0.6707	0.7419
TN_6	†	0.2607	0.6970	0.6975	0.3786	0.5207	0.6972	0.7626
POS_{10}		0.2149	0.6095	0.6611	0.3177	0.4456	0.6353	0.6861
WE_{wp}(concat)		0.2019	0.6055	0.6350	0.3025	0.4321	0.6203	0.6801
WE_{gn}(concat)		0.1959	0.6034	0.6226	0.2956	0.4258	0.6130	0.6434
WE_{wp}+WE_{gn}(concat)		0.2106	0.6235	0.6432	0.3146	0.4474	0.6334	0.6962
TN_{10}+POS_{10}		0.2647	0.5974	**0.7438**	0.3632	0.4724	0.6706	0.7407
TN_{10}+WE_{wp}(concat)	†	0.2634	0.7130	0.6948	0.3839	0.5302	0.7039	**0.7797**
TN_9+WE_{gn}(concat)	†	0.2552	0.7208	0.6778	0.3761	0.5267	0.6993	0.7638
TN_8+WE_{wp}+WE_{gn}(concat)	†	**0.2657**	**0.7309**	0.6911	**0.3890**	**0.5401**	**0.7110**	0.7772
POS_{10}+WE_{wp}(concat)		0.2174	0.6512	0.6405	0.3241	0.4619	0.6458	0.7120
POS_{10}+WE_{gn}(concat)		0.2081	0.6275	0.6356	0.3117	0.4455	0.6315	0.6956
POS_{10}+WE_{wp}+WE_{gn}(concat)		0.2199	0.6552	0.6462	0.3280	0.4670	0.6507	0.7202
TN_{10}+POS_{10}+WE_{wp}(concat)	†	0.2592	0.6931	0.6954	0.3760	0.5171	0.6942	0.7585
TN_{10}+POS_{10}+WE_{gn}(concat)	†	0.2474	0.6651	0.6863	0.3584	0.4937	0.6757	0.7472
TN_9+POS_{10}+WE_{wp}+WE_{gn}(concat)	†	0.2531	0.6850	0.6887	0.3679	0.5078	0.6868	0.7599

Table 3 presents the word embedding models that we test. For each model, we evaluate each of the composition functions presented in Section 3.3, *min*, *mean*, *max* and *concat*. As can be seen from Table 3, the models have 300 dimensional vectors and, hence, the functions *min*, *mean* and *max* result in 300 document features (900 for *concat*).

Classification and Metrics: For pre-processing and classification, we use scikit-learn[9]. As our classifier, we use SVM with a linear kernel and $C = 1.0$, since this theoretically motivated, default, parameter setting has been shown to provide the best effectiveness for text classification [2, 24]. We select F_2 as our main metric since sensitivity classification is a recall oriented task [3, 4], where the consequences of miss-classifying a sensitive document are much greater than miss-classifying a not-sensitive document. We also report the standard F-Measure (F_1) and, to account for class imbalance, we report Balanced Accuracy (BAC), where 0.5 BAC is random. We also report Precision, True Positive Rate (TPR), True Negative Rate (TNR) and the area under the Receiver Operating Characteristic curve (auROC) which, when documents are ranked by the output of a classifier's decision function, denotes the probability that a randomly selected positive instance is ranked higher than a randomly selected negative instance.

We test statistical significance, $p < 0.05$, using McNemar's non-parametric test [25] which is calculated from the prediction contingency tables for a pair of classifiers. Significant improvements compared to the text classification baseline (Text) are denoted with †. Additionally, in Table 5, significant improvements compared to the text classification with additional term features (Text+TN) are denoted with ‡.

5 Results

In this section, to answer the two research questions elicited in Section 4, we present the results of our classification experiments, over two tables: Table 4 presents the classification performance for each combination of *textual*, *grammatical* and *semantic* feature sets as *stand-alone* features; Table 5 presents the performance of each combination of feature sets extending the text classification baseline.

[9] http://scikit-learn.org/

Table 5. Results for combinations of *textual*, *grammatical* and *semantic* feature sets extending the text classification baseline.

Configuration		precision	TPR	TNR	F_1	F_2	BAC	auROC
Text		0.2410	0.6573	0.6841	0.3520	0.4874	0.6707	0.7419
Text+TN_9	†	0.2667	0.7010	0.7060	0.3858	0.5279	0.7035	0.7782
Text+POS_{10}	†	0.2596	0.6532	0.7160	0.3707	0.4999	0.6846	0.7498
Text+WE_{wp}(concat)	†	0.2474	0.6692	0.6905	0.3609	0.4984	0.6799	0.7584
Text+WE_{gn}(concat)	†	0.2435	0.6653	0.6850	0.3560	0.4933	0.6752	0.7459
Text+WE_{wp}+WE_{gn}(concat)	†	0.2557	0.6891	0.6947	0.3725	0.5138	0.6919	0.7594
Text+TN_6+POS_{10}	†	**0.2780**	0.6751	**0.7308**	0.3920	0.5224	0.7029	0.7725
Text+TN_9+WE_{wp}(concat)	†	0.2678	0.7090	0.7051	0.3881	0.5322	0.7070	**0.7874**
Text+TN_6+WE_{gn}(concat)	†	0.2699	0.7169	0.7044	0.3913	0.5371	0.7107	0.7784
Text+TN_7+WE_{wp}+WE_{gn}(concat)	† ‡	0.2730	**0.7229**	0.7069	**0.3956**	**0.5425**	**0.7149**	0.7859
Text+POS_{10}+WE_{wp}(concat)	†	0.2507	0.6493	0.7041	0.3609	0.4913	0.6767	0.7620
Text+POS_{10}+WE_{gn}(concat)	†	0.2515	0.6571	0.7020	0.3626	0.4950	0.6796	0.7546
Text+POS_{10}+WE_{wp}+WE_{gn}(concat)	†	0.2504	0.6532	0.7026	0.3612	0.4930	0.6779	0.7634
Text+TN_4+POS_{10}+WE_{wp}(concat)	†	0.2674	0.6811	0.7147	0.3827	0.5181	0.6979	0.7789
Text+TN_9+POS_{10}+WE_{gn}(concat)	†	0.2634	0.6830	0.7081	0.3786	0.5154	0.6955	0.7747
Text+TN_6+POS_{10}+WE_{wp}+WE_{gn}(concat)	†	0.2657	0.6910	0.7081	0.3825	0.5214	0.6995	0.7798

The baseline text classification approach (Text) is shown at the top of Tables 4 & 5, followed by sections for single, paired and triple feature sets respectively. We present results for term features (TN), grammatical features (POS) and semantic features (WE). For WE, we present the results of the single word embedding models, Wikipedia (WE_{wp}) and Google News (WE_{gn}), and when used together (WE_{wp}+WE_{gn}). Due to space constraints in Tables 4 & 5, we use F_2 as our preferred metric and present the best performing size of *n*-grams for TN and POS. For semantic features, we present the best performing composition function (*min*, *max*, *mean* or *concat*).

Firstly, we note that the text classification baseline (Text) achieves 0.4874 F_2 and 0.6707 BAC, markedly better than random (0.5 BAC). Addressing **RQ1**, from Table 4, we observe that semantic features (WE) on their own are competitive with, but do not out perform, the text classification baseline. Additionally, we can see that the *concat* composition function consistently performs best. These findings are in line with the findings of Balikas and Amini [19] on a different collection.

As single feature sets, only text *n*-gram features (TN) achieve significant improvements compared to the text classification baseline (0.5207 F_2 vs 0.4874 F_2), denoted as †. This shows that text features provide a strong foundation for sensitivity classification. Moreover, the best performing text *n*-gram size is $n = 6$, showing that larger sequences of text are indeed important for sensitivity classification. Adding semantic features to the text *n*-grams results in additional improvements, compared to the baseline, and TN_8+WE_{wp}+WE_{gn}(concat) achieves the best overall performance in Table 4.

From Table 5, we can see that extending text classification with semantic features significantly improves classification performance. The best performing configuration, Text+WE_{wp}+WE_{gn}(concat), achieves a 5.5% increase in F_2 score, compared with the baseline. However, extending text classification with term *n*-grams (Text+TN_9) achieves the best classification performance for single feature sets (+8.3% F_2).

Overall, the best performance is achieved when text classification is extended with additional *term* and *semantic* features combined, Text+TN_7+WE_{wp}+WE_{gn}(concat). This combination achieves 0.5425 F_2 and 0.7229 TPR, correctly classifying 9.99% more

sensitive documents than the text classification baseline. Notably, this combination also results in significant improvements compared to extending text classification with only term *n*-gram features (Text+TN_9), denoted as ‡ in Table 5.

In response to **RQ1**, firstly, we find that semantic word embedding features are, indeed, useful features for sensitivity classification. This is shown by the observation of significant improvements to classification effectiveness when they are added to the next best performing feature set, denoted by ‡ in Table 5. However, we conclude that the best overall classification performance is achieved when text classification is extended with additional *term n-gram* and *semantic* features. Moving to **RQ2**, Tables 4 & 5 show that using multiple embedding models, WE_{wp}+WE_{gn}, consistently out performs either of the single models, WE_{wp} or WE_{gn}, when they are used individually. Therefore, we conclude that using multiple word embedding models trained on different domains does, indeed, improve the effectiveness of semantic features for sensitivity classification.

6 Analysis

In this section, we provide analysis of the findings from our classification experiments. In Section 6.1, we discuss the classification predictions that are correct solely due to the word embedding features. In Section 6.2, we discuss the benefits for the sensitivity review process from extending text classification with semantic and term *n*-gram features.

6.1 Semantic Features

We now provide a short analysis of the documents we can correctly predict due to semantic features. We compare the best performing system, Text+TN+WE_{wp}+WE_{gn}, against text classification extended with term *n*-gram features, Text+TN.

Additional semantic features (from multiple domains) enable the classifier to convert 23 False Negative predictions to True Positive predictions, and 144 False Positive predictions to True Negative predictions. 13.77% of these converted predictions were sensitive documents. From the 23 converted sensitive documents, 15 are sensitive with respect to *International Relations*, 4 are sensitive with respect to *Personal Information* and 4 are sensitive with respect to both sensitivities.

Each of the documents with International Relations sensitivity contain multiple paragraphs that recount interactions and conversations between people and, moreover, the document's sensitivity is directly linked to these. This is in line with how we expect semantic features to enhance sensitivity classification, since these relations can be preserved in the dimensions of the vector representations. Interestingly, the sensitivities in documents relating to Personal Information also relate to actions, such as booking hotels, forced resignations and visa bans. Therefore, we intend to investigate such patterns of interaction relations further in future work, to develop classification rules for sensitivity and evaluate their cost/benefit trade-off for various sensitivity review user models.

6.2 Sensitivity Review

It is useful to provide sensitivity reviewers with a reliable way to predict how many sensitive documents remain in a partially reviewed collection. One way to approach this is to rank documents by a classifier's decision function output and review the ranking sequentially. We can then ask "how conservative does a classifier have to be, to correctly

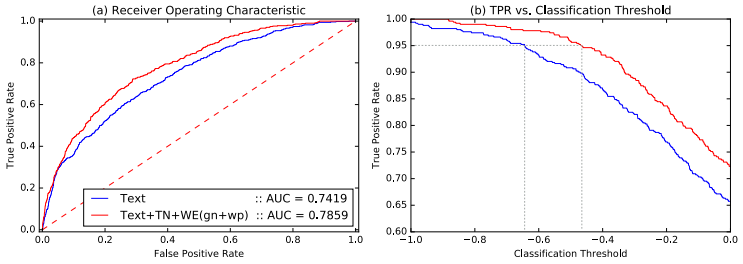

Fig. 2. (a) Receiver Operating Characteristic Curve. (b) True Positive Rate vs. Classification Threshold. The blue line shows the baseline text classification (Text) and the red line shows Text+TN_7+WE_{wp}+WE_{gn}(concat). The dashed line in (a) shows a random classifier. The dashed lines in (b) show the classification threshold required to achieve 0.95 TPR.

predict a certain percentage of sensitive documents?" In line with this user model, Figure 2 presents the Receiver Operating Characteristic curve, and True Positive Rate vs classification threshold for our classifier with additional term and semantic features, compared against the baseline text classification.

As can be seen from Figure 2(a), the additional features increase the True Positive Rate throughout the ranking. Therefore, a reviewer can have increased confidence in the system. Additionally, Figure 2(b), shows that semantic and term features enable the classifier to be less conservative. For example, the gray dashed lines in Figure 2(b) show that, with the additional features, we can correctly classify 95% of all sensitive documents by lowering the classification threshold to -0.46, whereas, the baseline would need to be set at -0.645. By using our approach, on this test collection, a reviewer would need to review 262 fewer documents to identify 95% of all sensitive documents.

7 Conclusions

In this work, we presented an effective approach for automatically classifying sensitive information in government documents, to assist the sensitivity review process. Our classifier deploys semantic features, derived from pre-trained word embedding models, to identify latent sensitive relations in documents. In a thorough evaluation, we compared the performance of the semantic features against grammatical and term features, as stand-alone features and extending text classification. We found that extending text classification with semantic features enabled our classifier to make significantly more accurate predictions, according to McNemar's test. Extending text classification with term *n*-gram and semantic features resulted in an 11.3% increase in F_2 score, correctly classifying 9.99% more sensitive documents than the baseline approach. Moreover, this approach markedly reduced the number of documents a reviewer would need to review to identify 95% of all sensitive documents in our collection (262 fewer documents).

Acknowledgements

The authors are thankful to the Foreign & Commonwealth Office and The National Archives of the UK for their support of this work.

References

1. DARPA: DARPA, New technologies to support declassification. (2010) http://fas.org/sgp/news/2010/09/darpa-declass.pdf.
2. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1) (2002) 1–47
3. McDonald, G., Macdonald, C., Ounis, I., Gollins, T.: Towards a classifier for digital sensitivity review. In: Proc. ECIR. (2014)
4. Berardi, G., Esuli, A., Macdonald, C., Ounis, I., Sebastiani, F.: Semi-automated text classification for sensitivity identification. In: Proc. CIKM. (2015)
5. Harris, Z.S.: Distributional structure. Word **10**(2-3) (1954) 146–162
6. Fung, B., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys (CSUR) **42**(4) (2010) 14
7. Fang, Y., Godavarthy, A., Lu, H.: A utility maximization framework for privacy preservation of user generated content. In: Proc. ICTIR. (2016)
8. Berardi, G., Esuli, A., Sebastiani, F.: A utility-theoretic ranking method for semi-automated text classification. In: Proc. SIGIR. (2012)
9. McDonald, G., Macdonald, C., Ounis, I.: Using part-of-speech n-grams for sensitive-text classification. In: Proc. ICTIR. (2015)
10. Lioma, C., Ounis, I.: Examining the Content Load of Part-of-Speech Blocks for Information Retrieval. In: Proc. COLING/ACL. (2006)
11. Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., Callison-Burch, C.: Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: Proc. ACL-IJCNLP. (2015)
12. Ghosh, D., Guo, W., Muresan, S.: Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In: Proc. EMNLP. (2015)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proc. NIPS. (2013)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proc. EMNLP. (2014)
15. Zheng, G., Callan, J.: Learning to reweight terms with distributed representations. In: Proc. SIGIR. (2015)
16. Zuccon, G., Koopman, B., Bruza, P., Azzopardi, L.: Integrating and evaluating neural word embeddings in information retrieval. In: Proc. ADCS. (2015)
17. Yang, X., Macdonald, C., Ounis, I.: Using word embeddings in twitter election classification. CoRR **abs/1606.07006** (2016)
18. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. CoRR **abs/1607.01759** (2016)
19. Balikas, G., Amini, M.: An empirical study on large scale text classification with skip-gram embeddings. CoRR **abs/1606.06623** (2016)
20. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive science **34**(8) (2010) 1388–1429
21. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297
22. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. JMLR **12** (2011) 2493–2537
23. Socher, R., Huang, E.H., Pennin, J., Manning, C.D., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Proc. NIPS. (2011)
24. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Springer (1998)
25. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika **12**(2) (1947) 153–157