RelDiff: Enriching Knowledge Graph Relation Representations for Sensitivity Classification

Hitarth Narvala[1], **Graham McDonald**[2], **Iadh Ounis**[2]
University of Glasgow, UK
[1]h.narvala.1@research.gla.ac.uk
[2]{graham.mcdonald,iadh.ounis}@glasgow.ac.uk

Abstract

The relationships that exist between entities can be a reliable indicator for classifying sensitive information, such as commercially sensitive information. For example, the relation *person-IsDirectorOf-company* can indicate whether an individual's salary should be considered as sensitive personal information. Representations of such relations are often learned using a knowledge graph to produce embeddings for relation types, generalised across different entity-pairs. However, a relation type may or may not correspond to a sensitivity depending on the entities that participate to the relation. Therefore, generalised relation embeddings are typically insufficient for classifying sensitive information. In this work, we propose a novel method for representing entities and relations within a single embedding to better capture the relationship between the entities. Moreover, we show that our proposed entity-relation-entity embedding approach can significantly improve (McNemar's test, $p < 0.05$) the effectiveness of sensitivity classification, compared to classification approaches that leverage relation embedding approaches from the literature (0.426 F_1 vs 0.413 F_1).

1 Introduction

More than a hundred countries have established Freedom of Information (FOI) regulations that require public organisations, such as governments, to release their official documents to the public (McDonald, 2019), for example the Freedom of Information Act 2000 in the UK.[1] Such regulations exempt the release of documents that contain *sensitive* information, for example personal or confidential information. Therefore, all government documents must be sensitivity reviewed to identify any potentially sensitive information before the documents can be considered for public release.

There is a growing need for automatic sensitivity classification approaches to assist government reviewers to sensitivity review large collections of digital documents, to comply with FOI laws (Prime and Russomanno, 2018). However, automatically classifying FOI sensitivities is a challenging task (McDonald et al., 2014), since sensitivity is often context-dependent. For example, information about an employee's salary details may, or may not, be sensitive depending on the role of the employee (e.g., a company director's salary may be in the public domain, whereas a regular employee's salary is usually considered to be personal information). Therefore, entities and the relations between entities can be an important indicator of sensitive information (Chakaravarthy et al., 2008).

We hypothesise that representing entity-relations in an embedding space can provide useful information for sensitivity classification and, in-turn, enable a sensitivity classifier to classify context-dependent sensitivities more effectively.

Studies such as (Rossi et al., 2021) showed that the relational information between entities in a knowledge graph can be effectively utilised to learn entity and relation embeddings. However, learning separate entity and relation embeddings may not be the most effective approach for sensitivity classification, since an entity or a relation alone is not a reliable indicator of sensitivity. This is illustrated in the example above, where the mention of a salary is potentially sensitive depending on whose salary is being discussed. Therefore, to capture the context of a potentially sensitive entity-relation, we argue that there is a need to capture the whole *entity-relation-entity* relationship (e.g., *person-isDirectorOf-company*) in a single embedding space.

In this work, we propose *RelDiff*: a novel approach for generating *entity-relation-entity* embeddings within a single embedding space. RelDiff adopts two fundamental vector algebraic operators to transform entity and relation embeddings from knowledge graphs into *entity-relation-entity* embeddings. We show that the RelDiff embeddings can be

[1]https://www.legislation.gov.uk/ukpga/2000/36/contents

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3671–3681

leveraged to improve the effectiveness of sensitivity classification. Moreover, we leverage six popular knowledge graph embedding (KGE) methods from the literature to compute RelDiff embeddings and compare the effectiveness of RelDiff against each of these KGE methods for sensitivity classification.

The contributions in this paper are three-folds: (1) we evaluate the importance of entity-relation embeddings for classifying sensitive information; (2) we propose RelDiff, a novel method to compose *entity-relation-entity* embeddings in a single embedding space using simple vector algebraic operations; and, (3) we show that our proposed RelDiff embedding features are significantly more effective for classifying sensitive information than knowledge graph embedding approaches from the literature.

To the best of our knowledge, this is the first work that effectively leverages entity-relation information for sensitivity classification. On a collection of government documents with real sensitivities (hereafter denoted as *GovSensitivity*), we show that integrating our RelDiff embeddings into sensitivity classification significantly improves (McNemar's test, $p < 0.05$) classification effectiveness, compared to several approaches from the literature that learn separate embeddings for entities and relations - e.g., RotatE (Sun et al., 2019) (F_1 0.426 vs 0.413).

2 Related Work

We now discuss related work on sensitivity classification and entity-relation representations.

Sensitivity Classification: The automatic classification of sensitive information, and protecting against the leakage of sensitive information from search systems, is an increasingly important topic that has received a lot of attention recently (McDonald and Oard, 2021; Olteanu et al., 2019b,a).

The task of automatically classifying FOI sensitivities[2] was first addressed by McDonald et al. (2014). The authors proposed to deploy separate classifiers with handcrafted features for specific FOI sensitivities ("Personal Information" and "International relations"). Differently from the work of McDonald et al. (2014), in this paper, we present a more advanced classifier by leveraging entity-relation information to classify sensitivities as a composite class of specific sensitivity types.

Another work by McDonald et al. (2017) evaluated various features for composite class sensitivity classification. McDonald et al. (2017) highlighted the effectiveness of semantic word embedding features and the sequence of document terms for sensitivity classification. Differently from the work of McDonald et al. (2017) we leverage entity-relation embeddings to effectively encode indicators of sensitivity and improve classification effectiveness.

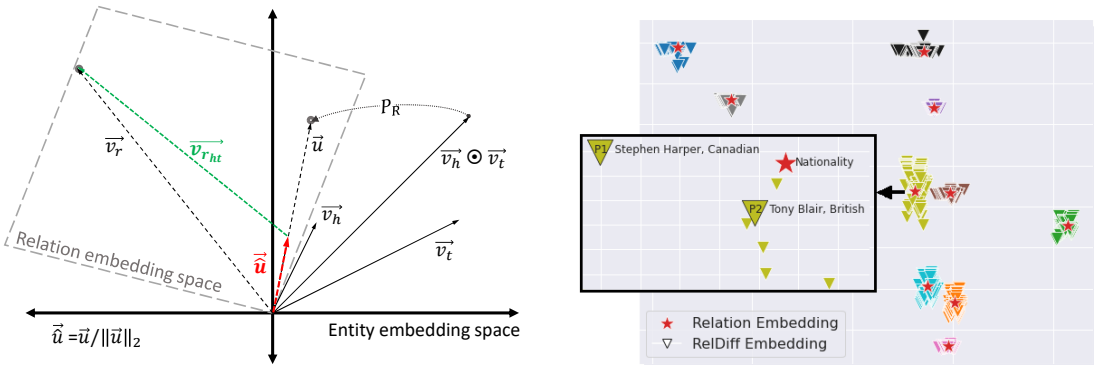
Previous studies have proposed to identify sensitive information using named entities. For example, Chakaravarthy et al. (2008) used a fixed database of public entities annotated to show which entities are sensitive, along with their associated predefined terms, to identify the context of sensitivities for document sanitisation. In contrast, Abril et al. (2011) considered all named entities as sensitive and utilised Named Entity Recognition (NER) to anonymise sensitive information. However, as described in Section 1, to identify context-dependent sensitivities where a majority of the entities are often not sensitive, we argue that entities themselves cannot indicate sensitivities reliably. Therefore, in this work, we propose an automatic approach to indicate whether the entities in a document constitute potential sensitive information by leveraging the relationship information between entities.

Berardi et al. (2015) and McDonald et al. (2020) have shown that sensitivity classification is indeed an effective approach for increasing the human efficiency of sensitivity review. Moreover, Sayed and Oard (2019) showed that increasing the effectiveness of sensitivity classification can also increase the retrieval effectiveness of sensitivity-aware IR systems. Differently from the work of Berardi et al. (2015); McDonald et al. (2020) and Sayed and Oard (2019), in this work, we further improve the effectiveness of sensitivity classification to better assist sensitivity reviewers.

Entity-Relation Representations: Various previous studies (Rossi et al., 2021; Ji et al., 2021) showed that knowledge graphs could be utilised to learn the representation of relationships between entities in an embedding space. We now provide a brief background of three popular categories of such knowledge graph embedding (KGE) methods as described by Rossi et al. (2021):

• Geometry-Based methods: they aim to model relationships as vector geometric operations such as translations (TransE - Bordes et al., 2013) or rotations (RotatE - Sun et al., 2019; HAKE - Zhang et al., 2020) in an embedding space. These methods work on the principle that if a relation r exists between the head and tail entities (h,t), then the vector for t should be similar to a vector obtained by operating h with r.

[2]https://www.legislation.gov.uk/ukpga/2000/36/part/II



(a) Computation of RelDiff vector $\vec{v}_{r_{ht}}$ using KGE relation vector $\vec{v}_r$ and entity vectors $\vec{v}_h$ & $\vec{v}_t$.

(b) RelDiff forms clusters of embeddings around the corresponding knowledge graph relation embedding.

Figure 1: Illustration of RelDiff Embeddings in 2d vector space.

• Tensor Factorisation-Based methods: these methods including RESCAL (Nickel et al., 2011) and TuckER (Balazevic et al., 2019) learn the relation representation by transforming all the h-r-t triples in a 3-dimensional binary tensor X, and then decompose the tensor X to compute the vectors of entities and relations.

• Neural Network-Based: these methods are becoming increasingly popular to represent knowledge graphs in a continuous neural features space. A number of methods have been proposed for learning relation representations by leveraging neural architectures such as methods based on Convolution Neural Networks (CNN) (ConvE - Dettmers et al., 2018; InteractE - Vashishth et al., 2020) and Graph Neural Networks (GNN) (R-GCN - Schlichtkrull et al., 2018; SACN - Shang et al., 2019) .

We evaluate recent *state-of-the-art* (SOTA) KGE methods from each of the above categories for sensitivity classification, namely: HAKE (Geometric), TuckER (Factorisation), InteractE (Neural CNN) and SACN (Neural GNN). In addition, we also evaluate the widely used TransE and RotatE methods. We provide further details of these methods in Section 3.2.

3 Entity-Relation Embeddings

In this section, we first present our proposed RelDiff approach for generating *entity-relation-entity* embeddings in Section 3.1. Second, in Section 3.2 we present knowledge graph embedding (KGE) approaches that we use as baselines for the evaluation of RelDiff.

3.1 Proposed Approach: RelDiff

Our proposed RelDiff approach generates *entity-relation-entity* embeddings in a single embedding space. Therefore, our approach can encode finer grained information about the relations than is captured when separate embeddings are learned for the entities and the relations, as is the case for the KGE approaches that we present in Section 3.2.

To construct our *entity-relation-entity* embeddings, we leverage two well-known vector algebraic operators for composing relational representations. First, we leverage the element-wise subtraction of a vector $\vec{v}_b$ from another vector $\vec{v}_a$ in an m-dimensional vector space $\mathbb{R}^m$, defined as:

$$\vec{v}_d = \vec{v}_a - \vec{v}_b \tag{1}$$

The resultant vector represents the direction from the vector $\vec{v}_b$ to the vector $\vec{v}_a$. Second, we leverage the element-wise multiplication (Hadamard product) of two vectors. Hadamard product has the effect of filtering and scaling shared features between two vectors and therefore can represent the mutual semantic composition between linguistic features such as words or sentences (Mitchell and Lapata, 2008) The Hadamard product ($\odot$) between two vectors is defined as:

$$\vec{v}_p = \vec{v}_a \odot \vec{v}_b \tag{2}$$

Our RelDiff method integrates the Subtraction and Multiplication operators using three vectors: (1) Head entity vector ($\vec{v}_h$), (2) Tail entity vector ($\vec{v}_t$) and (3) Relation vector ($\vec{v}_r$). We use the relation and entity vectors from the KGE approaches, presented in Section 3.2. In particular, we first perform Hadamard product (Equation 2) on $\vec{v}_h$ & $\vec{v}_t$ to obtain the semantic composition of the entity-pair. Due to the scaling effect, Hadamard product between the vectors of two entities can amplify the features that represent the relationship between the entities. For example, in the relation *UK-countryCaptial-London*, the Hadamard product of the embeddings for "UK" and "London"

can amplify the embedding dimensions that encode their geographical information. We then subtract the Hadamard entity-pair vector $\vec{v}_h \odot \vec{v}_t$ from the relation vector $\vec{v}_r$ using Equation 1 to compute the direction from the entity-pair vector to $\vec{v}_r$.

Different KGE models can represent entities and relations either in the same embedding space (e.g. TransE) or in separate embedding spaces (e.g. HAKE). However, to identify the direction from the entity-pair vector to the relation vector $\vec{v}_r$, the entity-pair vector and $\vec{v}_r$ are required in the same vector subspace. Therefore, before the subtraction operation, we project the entity-pair vector onto the relation embedding space $\mathbb{S}$ to effectively capture the direction from the entity-pair to the relation vector. To perform these projections, we prepare a projection matrix P_R for the relation embedding space $\mathbb{S}$ in three steps: (1) Find the basis vectors for $\mathbb{S}$ by performing Singular Value Decomposition on the relation embedding vectors. (2) Construct matrix A consisting of the basis vectors as columns. (3) Construct P_R using the following definition of the orthogonal projection matrix:

$$P_R = A.(A^t.A)^{-1}.A^t \quad (3)$$

where A^t is the transpose of A. To project the entity-pair vector onto $\mathbb{S}$ we perform a dot product of P_R with the entity-pair vector. During evaluations, we also found that it is beneficial to normalise the projected entity-pair vector with its L_2 norm. The RelDiff operation to produce a vector $\vec{v}_{r_{ht}}$ of a relation r corresponding to the entities (h & t) is illustrated in Figure 1(a), and is defined as follows:

$$\vec{v}_{r_{ht}} = \vec{v}_r - \vec{u}/||\vec{u}||_2, \text{ where } \vec{u} = P_R.(\vec{v}_h \odot \vec{v}_t) \quad (4)$$

Intuitively, the RelDiff operation can be explained as obtaining a vector pointing in the direction of the relation vector from another vector that is the semantic composition of the pair of related entities.

Figure 1(b) illustrates the RelDiff embeddings (denoted as $\bigtriangledown$) along with the relation embeddings that are produced by the KGE approaches (denoted as $\star$) in a 2-dimensional vector space. As shown in Figure 1(b), RelDiff clusters embeddings that share the same relation, but that have different related entities (the KGE relation embedding is the cluster centroid). Moreover, the similarity of the RelDiff embeddings for a particular *type of relation* is not affected by the low lexical similarity of the individual entities (Rogers et al., 2017) - i.e. "Stephen Harper" may not be similar to "Tony Blair". We expect this finer-grained representation of entity-relations to be beneficial for sensitivity classification, since the relation alone is not informative enough to be a reliable indicator of sensitivity.

3.2 Knowledge Graph Embeddings

As discussed in Section 2, a range of methods exist in the literature to learn embeddings of entities and relations that appear in a knowledge graph. The general idea behind learning entity-relation embeddings in such knowledge graph embedding methods (KGE) is as follows: given a relation r and its head-tail entities (h, t), optimise a scoring function $f_r(h, t)$. This function f_r can represent either or both of the following: (1) *Distance* between relational transformations of entities in a vector space (e.g. in the Geometric-Based methods). (2) *Semantic similarity* between entity-relation pairs (e.g. in the Neural Network-Based methods). We compare our proposed RelDiff approach against the following six KGE methods from the literature:

- TransE (Bordes et al., 2013) models a relation r as a translation in a vector space from head entity h to tail entity t, and optimises the distance between the translation vector $(h + r)$ and t.
- RotatE (Sun et al., 2019) extends TransE by leveraging a complex-vector space to model the relations as rotations from h to t.
- HAKE (Zhang et al., 2020) further extends RotatE by capturing a semantic hierarchy between the entities in a relation. For example, in the relation *UK-contains-Scotland*, "UK" is at a higher level of hierarchy than "Scotland".
- TuckER (Balazevic et al., 2019), leverages the tucker decomposition (Tucker, 1966) to compute entity and relation embeddings from a 3-dimensional tensor of the knowledge graph triples.
- InteractE (Vashishth et al., 2020) leverages a Convolution Neural Network (CNN) to model entity-relation embeddings by performing depthwise circular convolutions on different permutations of h and r.
- SACN (Shang et al., 2019) leverages both a CNN and a weighted Graph Convolution Network in learning relation embeddings by capturing structural information in a knowledge graph about the entity nodes and the strengths of the relation edges.

We deploy our proposed RelDiff approach using entity-relation embeddings from each of the aforementioned KGE approaches. To ensure a robust and fair comparison with the KGE approaches, we evaluate the effectiveness of RelDiff by comparing it to the following two methods:

KGRE: First, we use only the relation embeddings r from KGE as the features for sensitivity classification in order to evaluate the impact of generalised relation representations in identifying sensitivities.

CONCAT: Second, we concatenate the head-tail entity embeddings with the corresponding relation embedding, $concat(h, r, t)$, to compare the *entity-relation-entity* representations between KGE and RelDiff.

4 Classification Pipeline

In this section, we present our architecture pipeline for integrating entity-relation representations into sensitivity classification. The pipeline, illustrated in Figure 2, takes two inputs: a knowledge graph with pre-trained embeddings and the GovSensitivity collection containing sensitive and non-sensitive documents. The pipeline has five components: (1) The *Relation Extraction* component extracts entities and relations from the document collection and prepares a graph from the extracted relations. We present details about this component in Section 4.1. (2) The *Knowledge Graph Embedding* component deploys the KGE approaches we presented in Sections 3.2. (3) The *Relation Representation* component deploys the relation representation approaches that we presented in Sections 3.1 and 3.2 (RelDiff, KGRE and CONCAT). (4) The *Term Features* component constructs a bag-of-words representation of the GovSensitivity collection. (5) The *Sensitivity Classification* component trains the sensitivity classifier. We present the details of the Sensitivity Classification component in Section 4.2.

4.1 Relation Extraction

We leverage a relation extraction method from the literature (HRL-RE) presented by Takanobu et al. (2019) to jointly extract entities and relations in our GovSensitivity collection. HRL-RE is a hierarchical reinforcement learning method that deploys a tagging scheme to classify, firstly a relation mention in a text-span and secondly whether a token in the text-span participates to that relation.

To acquire the entity and relation embeddings for the GovSensitivity collection, we transform the extracted entity-relations into a graph structure where the nodes are the entities and the edges are the relations between the entities. We use this entity-relation graph of the GovSensitivity collection to train the KGE methods described in Section 3.2.

4.2 Sensitivity Classification

We deploy an ensemble classifier for sensitivity classification that combines two classifiers, i.e., a classifier that is trained on the bag-of-words document representations from the *Term Features* component of our pipeline and a second classifier that is trained on entity-relation embedding features (KGRE, CONCAT or RelDiff).

We choose to deploy an ensemble classifier for two reasons. First, the document features and the relation features are disjoint, i.e., they are independent without any direct correlation between the elements of each set. Therefore, a single classifier trained on both feature sets would likely miss specific statistical properties from each of the feature sets (Xu et al., 2013). Second, the term distribution-based document vectors are high dimensional and sparse, whereas relation embedding-based vectors are relatively low dimensional and dense. Hence, training separate classifiers can more effectively capture the specific characteristics of the individual feature sets (Sun, 2013).

For our ensemble approach, as shown in Figure 2, we deploy a stacking ensemble (Wolpert, 1992) with two classifiers E_{Txt} & E_{Rel} that are trained using document term features and relation embedding features, respectively. To combine the classifiers' outputs, we normalise the confidence scores from E_{Txt} & E_{Rel} using L_2 norm, and concatenate the normalised scores S_{Txt} & S_{Rel} as two features to train a meta-classifier E_M for sensitivity classification.

In E_{Rel}, we construct the document representation for a given document d by aggregating the *entity-relation-entity* embeddings (or relation embeddings in KGRE configuration) of all the relations in d. We utilise the element-wise mean operation for aggregating the embedding vectors $x \in R_d$ (where R_d is an m-dimensional embedding subspace), i.e., the document representation for the i^{th} dimension d_i is defined as:

$$d_i = \underset{x \in R_d}{\text{mean}}(x_i) \quad \forall\, i \in [0, m-1] \tag{5}$$

5 Experimental Setup

In this work, we address the following two research questions:

- **RQ1**: Does integrating knowledge graph embeddings into sensitivity classification help to more effectively classify context-dependent sensitivities?
- **RQ2**: Are RelDiff *entity-relation-entity* embeddings more effective for sensitivity classification than learning separate entity & relation embeddings?

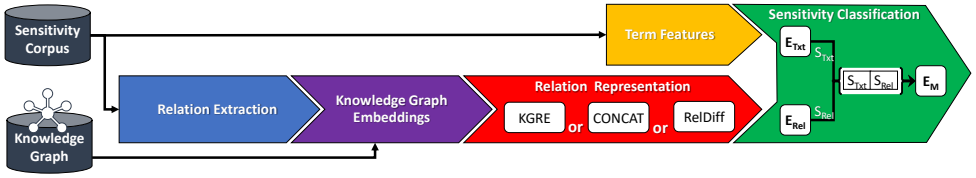


Figure 2: Pipeline for integrating entity-relations into Sensitivity Classification.

Sensitivity Collection: We use a collection of 3801 government documents (GovSensitivity), as our main dataset for sensitivity classification. GovSensitivity contains 502 sensitive and 3299 non-sensitive documents that are reviewed by government sensitivity reviewers to identify two FOIA sensitivities, i.e., "Personal Information" and information impacting "International Relations". We use stratified sampling to split this collection into train, validation, and test datasets across 5-folds to perform Cross Validation.

Baselines: In addition to evaluating our proposed RelDiff approach against the KGE approaches that we presented in Section 3.2 (KGRE & CONCAT), we also report the effectiveness of two baseline sensitivity classifiers. First, we report the effectiveness of an SVM classifier with a linear kernel and the regularisation parameter set as $C = 10$. The parameters C represents the strength of L_2 regularisation penalty. This approach, denoted as *TC* in Section 6, is trained on TF-IDF n-grams term features, where we set $n <= 4$ through grid search on validation dataset in the range $n \in [1, 4]$. The second baseline sensitivity classifier that we report is identical to the TC baseline classifier, except that this classifier, denoted as *TC-Enrich* in Section 6, is trained on an enriched version of the GovSensitivity collection, where the documents have been enriched (Bryl et al., 2010; Pantel and Fuxman, 2011) by adding a *relation token*, e.g., "*place_of_birth*", for each of the extracted entity-relations.

Relation Extraction: For relation extraction, we train the relation extraction model HRL-RE[3] (Takanobu et al., 2019) on NYT10 dataset (Riedel et al., 2010). Before extracting relations from the GovSensitivity collection, we remove the header section of the documents and split the documents into sentences using the spaCy (Honnibal et al., 2020) language model *en_core_web_lg*. HRL-RE extracted 46,610 entity-relation triples, for 23,609 unique entities and 18 relation types in the GovSensitivity collection. We transform the extracted entity-relations into a graph structure corresponding to each fold of the GovSensitivity collection. Table 1 shows the average number of entities, relations and entity-relation triples across each fold of the GovSensitivity collection.

Table 1: Number of entities, relations and observed triples in GovSensitivity compared to Freebase.

Dataset	#entities	#relations	#triples
GovSensitivity	10,495	18	21,632
FB15k-237	14,541	237	310,116

Table 2: Results for link prediction on GovSensitivity.

	MRR	H@10		MRR	H@10
TransE	0.369	0.528	TuckER	**0.468**	0.535
RotatE	0.436	**0.561**	InteractE	0.198	0.251
HAKE	0.453	0.553	SACN	0.281	0.426

Knowledge Graph Embeddings: As shown in Table 1, the GovSensitivity graph is relatively small as compared to popular Knowledge Graphs such as Freebase (Bollacker et al., 2008). Therefore, we deploy a transfer-learning approach to train the KGE methods TransE, RotatE, HAKE, TuckER, InteractE and SACN. First, we pre-train the aforementioned KGE methods on the FB15K237 subgraph of Freebase, each using their publicly available implementations and the best hyperparameters specified in the respective papers. Second, we train the pre-trained KGE models separately on each fold of the GovSensitivity graph. Table 2 presents the link prediction results on the GovSensitivity collection graph in terms of Mean Reciprocal Rank (MRR) and Hits@10 (H@10).

RelDiff Embeddings: For computing the RelDiff embeddings, we leverage the entity and relation embeddings trained on the GovSensitivity Collection graph from the KGE approaches, TransE, RotatE, TuckER, InteractE and SACN.

Sensitivity Classification: As we previously discussed in Section 4.2, we deploy an ensemble classification approach to integrate entity-relation embeddings into sensitivity classification. For the ensemble classifier, as illustrated in Figure 2, we deploy E_{Txt} as the baseline text classifier (TC), E_{Rel} as an SVM classifier with a linear kernel and the meta-classifier E_M as a Logistic Regression

[3]We use the following implementation for HRL-RE: https://github.com/truthless11/HRL-RE

Table 3: The evaluated configurations for sensitivity classification. ($m \in$ {TransE, RotatE, HAKE, TuckER, InteractE, SACN})

Identifier	Description
TC	Baseline SVM text classifier with bag-of-words (BoW) term features.
TC-Enrich	SVM text classifier comprising BoW from enriched documents.
KGRE$_m$	Ensemble classifier (EC) with BoW & relation embeddings from m.
CONCAT$_m$	EC with BoW & concatenated entity-relation embeddings from m.
RelDiff$_m$	EC with BoW & RelDiff entity-relation embeddings from m.

classifier. The regularisation parameter for both E_{Rel} and E_M is set using grid search on a validation dataset in the range $C \in \{10^x \; \forall \; x \in [-5, 4]\}$.

To test for statistical significance, we use McNemar's non-parametric test (McNemar, 1947) with a significance threshold $p < 0.05$.

6 Results

In this section, we present the results of our sensitivity classification experiments. Table 3 presents the evaluated classifiers and the notations that we use to refer to them hereafter. Table 4 presents the classification results in terms of precision (prec), recall, F_1 and balanced accuracy (BAC). In Table 4, the evaluated classifiers under different KGE configurations are shown, e.g., RelDiff$_{\text{TransE}}$ represents the classifier with RelDiff embeddings computed using the TransE entity-relation embeddings. Additionally in Table 4, significant improvements compared to the baseline text classifier (TC), the KGRE and the CONCAT configurations of the ensemble classifiers are denoted with $*$, † and ‡, respectively.

First, addressing **RQ1**, we observe from Table 4 that the entity-relation embeddings features in the KGRE and RelDiff configurations of the ensemble classifiers significantly improve the effectiveness of sensitivity classification, compared to the baseline text classifier TC ($p < 0.05$, denoted as $*$), e.g. BAC 0.739 RelDiff$_{\text{RotatE}}$ & 0.730 KGRE$_{\text{RotatE}}$ vs 0.728 TC. The improvements are significant consistently across all six configurations (TransE, RotatE, HAKE, TuckER, InteractE, SACN) for RelDiff and across four of the KGE configurations (TransE, RotatE, InteractE, SACN) for KGRE. Sensitivity classification on documents enriched with entity-relation tokens (TC-Enrich) shows a similar performance (0.730 BAC) to KGRE. However, RelDiff outperforms TC-Enrich across all six configurations. Therefore, in response to **RQ1**, we conclude that representing entity-relations in an embedding space does indeed significantly improve the effectiveness of entity-relations for sensitivity classification.

Table 4: Results for combinations of KG embeddings (KGRE/CONCAT) and RelDiff embeddings compared with a baseline text classification and document enrichment.

Configuration		prec	recall	F_1	BAC
TC		0.282	0.745	0.409	0.728
TC-Enrich	$*$	0.280	0.755	0.409	0.730
KGRE$_{\text{TransE}}$	$*$	0.287	0.741	0.414	0.730
CONCAT$_{\text{TransE}}$		0.232	0.773	0.357	0.692
RelDiff$_{\text{TransE}}$	$*$ ‡	0.287	0.745	0.415	0.732
KGRE$_{\text{RotatE}}$	$*$	0.287	0.741	0.413	0.730
CONCAT$_{\text{RotatE}}$		0.284	0.745	0.412	0.730
RelDiff$_{\text{RotatE}}$	$*$†‡	**0.298**	0.745	**0.426**	**0.739**
KGRE$_{\text{HAKE}}$		0.285	0.743	0.412	0.730
CONCAT$_{\text{HAKE}}$		0.285	0.743	0.412	0.730
RelDiff$_{\text{HAKE}}$	$*$†‡	0.290	0.747	0.418	0.735
KGRE$_{\text{TuckER}}$		0.285	0.743	0.412	0.730
CONCAT$_{\text{TuckER}}$		0.230	0.733	0.350	0.680
RelDiff$_{\text{TuckER}}$	$*$ ‡	0.290	0.749	0.418	0.735
KGRE$_{\text{InteractE}}$	$*$	0.284	0.741	0.411	0.728
CONCAT$_{\text{InteractE}}$	$*$	0.284	0.741	0.411	0.728
RelDiff$_{\text{InteractE}}$	$*$	0.286	0.745	0.413	0.731
KGRE$_{\text{SACN}}$	$*$	0.279	0.755	0.408	0.729
CONCAT$_{\text{SACN}}$	$*$	0.279	0.755	0.408	0.729
RelDiff$_{\text{SACN}}$	$*$	0.282	**0.763**	0.412	0.734

To address **RQ2**, we evaluate the effectiveness of sensitivity classification when leveraging the RelDiff *entity-relation-entity* embeddings compared to leveraging the entity and relation embeddings from the KGE approaches (KGRE & CONCAT). First, we note that the ensemble classifier with RelDiff embeddings achieves the best overall sensitivity classification performance in terms of F_1 (0.426) , BAC (0.736) and precision (0.298) (for the RotatE configuration) and recall (0.763 for the SACN configuration). Moreover, RelDiff results in significantly improved sensitivity classification effectiveness ($p < 0.05$, denoted as †) compared with KGRE for two configurations (RotatE and HAKE) and compared with CONCAT for four configurations (TransE, RotatE, HAKE and TuckER) ($p < 0.05$, denoted as ‡).

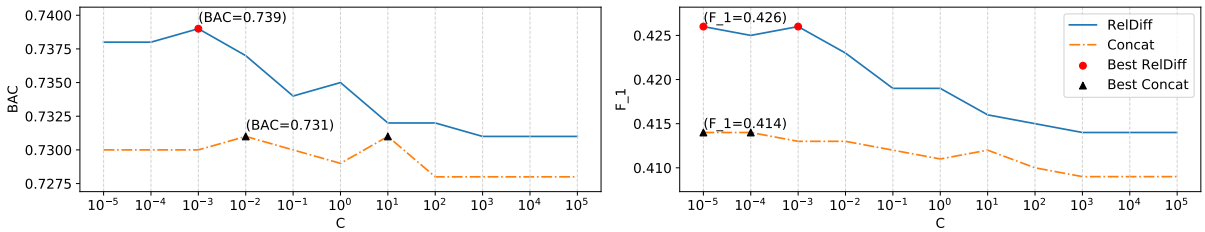


Figure 3: Effect of regularisation in the ensemble meta-classifier on BAC and F_1.

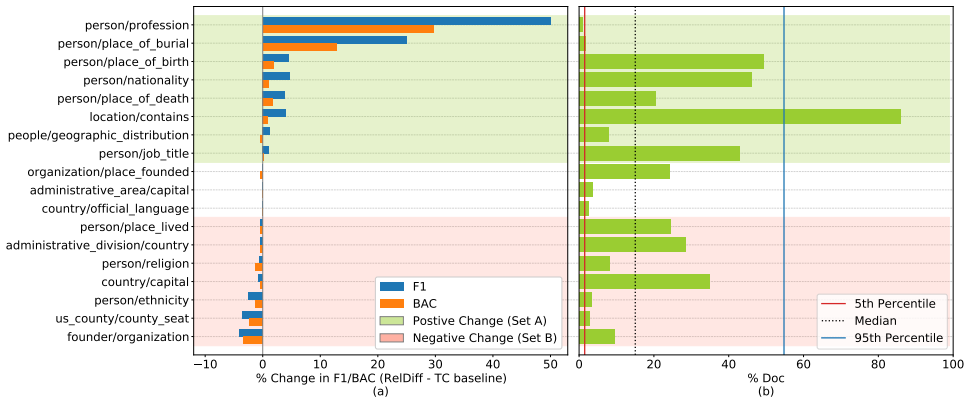


Figure 4: Improvements in F_1 and BAC by $\text{RelDiff}_{\text{RotatE}}$ as compared to the TC baseline with respect to different relation types.

We also note that, except for the RotatE configuration, both the KGRE and CONCAT ensemble classifiers achieve either a lower precision or recall as compared to the TC baseline. Whereas, the RelDiff ensemble classifiers often outperform TC across all four metrics, and are still competitive otherwise. Lastly, we note that the CONCAT ensemble classifiers show similar performances to KGRE in most configurations, and achieves lowest performances for TransE and TuckER configurations. Therefore, in response to **RQ2**, we conclude that our proposed RelDiff approach for generating *entity-relation-entity* embeddings does indeed lead to significant improvements in sensitivity classification effectiveness as compared to TC, KGRE and CONCAT. We also conclude that a concatenation of entity and relation embeddings (CONCAT) does not provide effective *entity-relation-entity* embeddings for sensitivity classification.

7 Analysis

We now provide an analysis of the findings from our experiments. We discuss the effect of regularisation in the ensemble classifiers in Section 7.1. In Section 7.2, we describe the contribution of individual relation types on the effectiveness of sensitivity classification. In Section 7.3, we discuss the importance of the improvements by RelDiff in sensitivity classification effectiveness for sensitivity review.

7.1 Effect of Regularisation

For ensemble learning classifiers, we provide a short analysis of the effect of the regularisation parameter C in the ensemble's meta-classifier (E_M from Figure 2) on the sensitivity classification performance. To do this, we keep the regularisation parameters of the first-layer classifiers (E_{Txt} & E_{Rel}) fixed and plot the overall classification BAC and F_1 for different values of the meta-classifier's regularisation parameter C. Figure 3 illustrates the variation in performances of the $\text{RelDiff}_{\text{RotatE}}$ and $\text{CONCAT}_{\text{RotatE}}$ ensemble classifiers as the regularisation of the meta-classifier is varied. As we can see from Figure 3, both RelDiff and CONCAT ensemble classifiers usually perform better at lower values of C, and the classifiers' performance gradually degrades for higher values of C. However, the CONCAT classifier never outperforms the RelDiff classifier. This observation provides further evidence to support our answer to **RQ2**, namely that RelDiff provides significantly more effective entity-relation representations than the KGE approaches for sensitivity classification.

7.2 Contribution of Different Relation Types

It is also useful to analyse the contribution of the individual relation types on the effectiveness of sensitivity classification. Figure 4(a) illustrates the F_1/BAC improvements from the RelDiff

ensemble classifier, compared to the TC baseline, for documents containing each of the relation types. Figure 4(b) shows the frequency of documents in the GovSensitivity collection with respect to the relations they contain. Overall, we note that not all relations improve F_1 and BAC. For example, the person-entity-relations *place_of_birth* and *nationality* improve F_1 by 4.50% and 4.75%, respectively in RelDiff as compared to the TC baseline, whereas the relations *us_county/county_seat* and *founder/organisation* degrade F_1 in RelDiff by 2.60% and 3.53%, respectively. Out of a total of 18 relations types, RelDiff improves the F_1 metric for 8 relations (Figure 4(a) Set A), while it obtains lower F_1 scores for 7 relations (Figure 4(a) Set B). However, from Figure 4(b), we note that the document frequency for the relations in Set A is notably higher as compared to the relations in Set B (e.g. 49.3% for *place_of_birth* vs 9.85% for *founder/organisation*). This comparison of classification improvements together with document frequency clearly shows that RelDiff can improve sensitivity classification for the relation types that appears more frequently in the GovSensitivity collection. We also observe that RelDiff improves the F_1 metric for 7 out of 10 person-entity relations types. This further shows that RelDiff can effectively identify personal sensitive information. Overall, the above analysis indicates that various entity-relation types, and the number of documents that they appear in, can affect the effectiveness of sensitivity classifiers that leverage entity-relations. We will investigate this further as future work.

7.3 Importance to Sensitivity Review

When assisting sensitivity reviewers with sensitivity classification predictions, there can be a substantial difference in reviewing speeds for False Positive (FP) (non-sensitive document predicted as sensitive) and True Negative (TN) predictions (McDonald et al., 2020). Compared to the TC baseline, $\text{RelDiff}_{\text{RotateE}}$ converts 77 FPs to TNs (8.03%) on our collection (mean document length=1066.78). McDonald et al. (2020) reports a 53% increase in reviewing speeds for TN predictions compared to FPs (288.13 wpm vs 188.38 wpm). Based on these reviewing times, the converted documents would take "4.75 hours" to review using $\text{RelDiff}_{\text{RotateE}}$ compared to "7.27 hours" for the TC baseline. Therefore, the improvements shown by RelDiff can markedly reduce the amount of time required to sensitivity review a collection of documents. This is an important contribution that will assist the governments in meeting their legal obligations to publicly release their documents in a timely manner. Moreover, going forward, as the sizes of the collections that must be sensitivity reviewed increase, the benefits to governments from these reduced reviewing times will grow markedly larger.

8 Conclusions

We proposed a method, RelDiff, to represent *entity-relation-entity* triples in an embedding space for automatic sensitivity classification. We compared the RelDiff embedding features with embeddings from popular and SOTA knowledge graph methods (KGE) and term features from documents enriched with entity-relations. In general, all relation representation methods we evaluated, consistently improved the effectiveness of sensitivity classification over baseline text classifiers. However, we showed that the KGE methods are insufficient to effectively represent entity-relation information for sensitivity classification. On the other hand, our proposed approach RelDiff can leverage these existing KGE methods to produce an effective entity-relation representation for sensitivity classification. From the different configurations of KGE methods, we found that the RelDiff features can significantly improve the performance of sensitivity classification (0.739 BAC & 0.426 F_1, $\text{RelDiff}_{\text{RotatE}}$) in comparison to a baseline text classifier (0.728 BAC & 0.409 F_1) and KGE baselines (0.730 BAC & 0.412 F_1, $\text{CONCAT}_{\text{RotatE}}$), according to the McNemar's test, $p < 0.05$. Moreover, while the overall classification performance varies according to the L_2 regularisation penalty in an ensemble classifier, the classifier with the KGE features never outperforms the classifier with the RelDiff features. Furthermore, since false positive (FP) predictions can affect the speed of sensitivity reviewers (McDonald et al., 2020), RelDiff classifiers can markedly increase the sensitivity reviewers' speed due to the notably lower FPs compared to the text classification baseline (up to 53% speed gain for 8.03% documents). We also showed that various relation types, such as *person/place_of_birth* and *founder/organisation*, have different effects on the classification performance. We will investigate this important and interesting research as future work.

References

Daniel Abril, Guillermo Navarro-Arribas, and Vicenç Torra. 2011. On the declassification of confidential documents. In *Modeling Decision for Artificial Intelligence*, pages 235–246.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194.

Giacomo Berardi, Andrea Esuli, Craig Macdonald, Iadh Ounis, and Fabrizio Sebastiani. 2015. Semi-automated text classification for sensitivity identification. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, page 1711–1714.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, page 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, page 2787–2795.

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. Using background knowledge to support coreference resolution. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 759–764.

Venkatesan T Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K Mohania. 2008. Efficient techniques for document sanitization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, page 843–852.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1811–1818.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

G. McDonald, C. Macdonald, and I. Ounis. 2017. Enhancing sensitivity classification with semantic features using word embeddings. In *Proceedings of the 39th European Conference on Information Retrieval*, pages 450–463.

Graham McDonald. 2019. *A framework for technology-assisted sensitivity review: Using sensitivity classification to prioritise documents for review*. Ph.D. thesis, University of Glasgow.

Graham McDonald, Craig Macdonald, and Iadh Ounis. 2020. How the accuracy and confidence of sensitivity classification affects digital sensitivity review. *ACM Transactions on Information Systems*, 39(1).

Graham McDonald, Craig Macdonald, Iadh Ounis, and Timothy Gollins. 2014. Towards a classifier for digital sensitivity review. In *Proceedings of the 36th European Conference on Information Retrieval*, pages 500–506.

Graham McDonald and Douglas W Oard. 2021. Search among sensitive content. ECIR 2021 tutorial.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 236–244.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*.

Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, and Michael D. Ekstrand. 2019a. Workshop on fairness, accountability, confidentiality, transparency, and safety in information retrieval (FACTS-IR). In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1423–1425.

Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, Michael D. Ekstrand, Adam Roegiest, Aldo Lipani, Alex Beutel, Ana Lucic, Ana-Andreea Stoica, Anubrata Das, Asia Biega, Bart Voorn, Claudia Hauff, Damiano Spina, David D. Lewis, Douglas W. Oard, Emine Yilmaz, Faegheh Hasibi, Gabriella Kazai, Graham McDonald, Hinda Haned, Iadh Ounis, Ilse van der Linden, Joris Baan, Kamuela N. Lau, Krisztian Balog, Mahmoud F. Sayed, Maria Panteli, Mark Sanderson, Matthew Lease, Preethi Lahoti, and Toshihiro Kamishima. 2019b. FACTS-IR: fairness, accountability, confidentiality, transparency, and safety in information retrieval. *SIGIR Forum*, 53(2):20–43.

Patrick Pantel and Ariel Fuxman. 2011. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 83–92.

Tyler Prime and Joseph Russomanno. 2018. The future of FOIA: Course corrections for the digital age. *Communication law and policy*, 23(3):267–300.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 148–163.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2).

Mahmoud F. Sayed and Douglas W. Oard. 2019. Jointly modeling relevance and sensitivity for search among sensitive content. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 615–624.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Procedings of the 15th Extended Semantic Web Conference*, pages 593–607.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3060–3067.

Shiliang Sun. 2013. A survey of multi-view machine learning. *Neural computing and applications*, 23(7):2031–2038.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7072–7079.

Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3009–3016.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv:1304.5634*.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3065–3072.