A Self-Training Ontology-Driven Approach for Topic Classification (ST-OLDA)

Qi Hao, Jeroen Keppens, and Odinaldo Rodrigues

***Abstract*—Latent Dirichlet Allocation (LDA) is a topic classification technique that produces a probabilistic model based on word co-occurrence, for the purpose of text classification. Conventional LDA ignores the fact that words may have multiple meanings and that different words may have the same meaning. This focus on the words rather than their meanings limits the accuracy of the classification. This work introduces an intermediate labelling component to LDA using the concepts in DBpedia's ontology to help capture some of the possible meanings of the words appearing in the documents. We call this novel technique Ontology-Driven LDA (OLDA). As for LDA, OLDA can be combined with a self-training procedure to reduce the amount of manually classified data required (we refer to the self-training variant as ST-OLDA). We compared the classification performance of ST-OLDA against the performance of two other leading self-training classification methods: ST Term Frequency-Inverse Document Frequency (ST TF-IDF) and ST-LDA. Our experimental results show that the inclusion of the ontology component helps to reduce the training time by nearly half whilst achieving the highest accuracy in the classification of four widely used datasets. In particular, ST-OLDA outperforms ST-LDA's accuracy of classification by as much as 11%.**

***Index Terms*—NLP, topic classification, LDA, semi-supervised learning**

I. Introduction

Recent advances in natural language processing enable the collection and analysis of unstructured text data with an unprecedented breadth and scale [1]. An important analysis task is the classification of documents into topics or categories, so that humans can discover texts that are of interest to them more easily. Obviously, as manual topic extraction is time-consuming and does not scale well, automating this process is very important and has been done successfully in some specific domains, such as Tweet classification [2], public opinion monitoring [3], personalised recommendation systems [4], as well as legal documents [5].

Many existing classification techniques can summarise text into topics (*topic modelling*) and accordingly identify topic terms and classify texts (*topic classification*). Latent Dirichlet Allocation (LDA) is one of the most commonly used *topic modelling* techniques [2], [3], [6]. LDA is a probabilistic model that projects a document into the topic space using the Dirichlet probability distribution [7]. Each topic is seen as a collection of words and their probability distribution [8]. An LDA model can be produced by both supervised and unsupervised machine learning techniques. In general, supervised techniques produce LDA models that vastly outperform those produced by unsupervised techniques [2], [3], [6]. However, supervised techniques need a training dataset that is manually generated and very costly to produce [9], and as a result these training datasets are usually small. However, larger training datasets not only assure better generalisation, but also provide better accuracy. In order to overcome the cost of obtaining a large training dataset, [10] suggested the introduction of a self-training phase to automatically enlarge an initially small amount of training data. [11] used this idea with LDA resulting in a technique called *Self-Training LDA* (ST-LDA). Once the enlarged training dataset is generated, the topic classification can then be performed using a conventional supervised technique, such as Support Vector Machine (SVM).

It is also possible to incorporate expert knowledge during the generation of the topic model itself [12] instead of the use of a self-training phase. However, this requires a larger amount of human effort [11].

As conventional LDA approaches use words as self-contained tokens, they ignore the fact that words may have multiple meanings and that different words may have the same meaning, thereby limiting the potential accuracy of the models they produce [13]. For example, the sentences "Google is launching their new phone" and "Microsoft is stepping into the study of advanced electronics" would not be classified into the same category using LDA because they do not have any relevant words in common. However, these sentences *are* related because Microsoft and Google are both "companies" involved with "technology". Our idea is to bridge this gap by looking at some concepts associated with the words "Microsoft" and "Google". This can be done by making use of a database containing a good set of cross-domain knowledge such as DBpedia or WordNet. In this work, we chose DBpedia, because it contains much more general knowledge than WordNet.

DBpedia contains structured content about over 6.0 million entities, classified in a consistent ontology. In DBpedia each word is associated with a set of labels describing its general properties within the ontology. Our approach uses these labels to find implicit relationships between words and therefore increase the overall accuracy of the classification. In our previous example, Microsoft and Google would be associated through the labels "company" and "technology" that they share in DBpedia. Including this ontological knowledge as an

Qi Hao is with the Department of Informatics, King's College London, London, UK (e-mail: qi.hao@kcl.ac.uk).

Jeroen Keppens is with the Department of Informatics, King's College London, London, UK (e-mail: jeroen.keppens@kcl.ac.uk).

Odinaldo Rodrigues is with the Department of Informatics, King's College London, London, UK (e-mail: odinaldo.rodrigues@kcl.ac.uk).

intermediate labelling component in LDA has the following advantages: (i) it allows the topics to be defined more generally in terms of ontological concepts rather than words and this captures the semantical meaning of the words more accurately; (ii) as a side-effect, we will see that this extra dimension helps to reduce the training and classification times. In virtue of the use of this ontological knowledge, we call the resulting technique *Ontology-Driven Latent Dirichlet Allocation* (OLDA).

As for LDA, OLDA can also employ a self-training phase in order to enlarge the initial amount of manually classified data. Accordingly, we call the variant using the self-training phase *Self-Training Ontology-Driven Latent Dirichlet Allocation* (ST-OLDA). The self-training can be performed with any appropriate procedure. We considered two alternatives: a relatively ad hoc method employing a logistic regression model and the procedure proposed in [11], which employs Gibbs sampling. The former is faster to train but its classification is less accurate. In our experiments, the combination of the more precise self-training technique with OLDA outperformed ST-LDA by as much as 11.01% (in the R52 dataset) and even the classifier proposed by [14] (which is not self-trained and requires more training data) in the "20 Newsgroups" dataset by nearly 8%.

The remainder of this paper is organised as follows. Section II provides some background about topic modelling. Section III presents our new OLDA approach. Section IV describes the two self-training techniques and how they can be incorporated with OLDA. Section V describes the results of our experimental analysis and Section VI concludes with a discussion and areas for future work.

II. Background

Topic modelling is a type of statistical modelling for discovering the abstract topics that occur in a collection of documents. Intuitively speaking, given that a document is about a particular topic, one would expect certain words to appear in the document more frequently, and others less frequently. The "topics" produced by topic modelling techniques are collections of related words. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents to discover, based on the statistics of the words in each one, what the topics might be and what each document's balance of topics is.

Topic extraction approaches commonly use a Vector Space Model (VSM) representation, where topics are based on words as independent units and are weighed using Term-Frequency - Inverse Document Frequency (TF-IDF) [15]. However, the sole use of word frequency is not sufficient to differentiate between topics in many situations where the order of the appearance of the word is important [16]. To address this, Nigam et al. employs a Naïve Bayes classifier to perform parameter estimation of a statistical Expectation Maximisation (EM) model (EM-NB) [17]. A significant drawback of this approach is that it converges to a local optimum. Deerwester et. al proposed Latent Semantic Analysis (LSA) to represent text as latent concepts in a low dimensional semantic space [18]. LSA was further enhanced by interpreting topics as multinomial word distributions using the probability density function, yielding a technique called Probabilistic Latent Semantic Analysis (PLSA) [19]. However, PLSA is prone to overfitting, i.e., incorrectly classifying documents. In order to address PLSA's overfitting problems, Blei et al. proposed an improved model based on the Dirichlet prior probability distribution [8] – the so-called Latent Dirichlet Allocation (LDA). Each document in LDA is represented as a multinomial distribution of topics, where topics are clusters of words.

Fig. 1 depicts a typical LDA matrix representation. Let $\mathscr{D} = \{D_1, \ldots, D_n\}$ be a collection of documents to classify into the topics $\mathscr{T} = \{T_1, \ldots, T_l\}$ and $\mathscr{W} = \{W_1, \ldots, W_k\}$ be the set of words appearing in $\mathscr{D}$. Δ is the matrix associating documents to words, where each d_{ij} is 1 if the word $W_j \in \mathscr{W}$ appears in the document $D_i \in \mathscr{D}$ and 0 otherwise. LDA treats each document in a collection as having been created from several latent topics, each of which having an associated probability distribution of co-occurring words [13]. This probability distribution is captured by a $l \times k$ matrix Φ with the probabilities p_{ab} of each topic T_a ($1 \leq a \leq l$) being described by word W_b ($1 \leq b \leq k$). The LDA model aims to obtain from Δ and Φ a $n \times l$ documents/topics matrix Θ with the probabilities q_{xy} of each document D_x ($1 \leq x \leq n$) being associated with topic T_y ($1 \leq y \leq l$).

In spite of its strengths, LDA sometimes fails to capture the true semantical meaning of the topics. Word-assignment ambiguity, homonyms and polysemous words can introduce noise into the model [20] which some variations of the method have attempted to reduce. Panichella et al. used a genetic algorithm to fine-tune the prior probabilities in Φ and Θ [21]. Hsu et al. proposed a supervised hybrid LDA approach utilising a genetic algorithm to optimise the weight vector of the documents-topics matrix Θ [22]. Krasnashchok et al. employed named entities to recognise domain-specific terms and introduced a new weighting model, improving interpretability, specificity and the diversity of the extracted topics [23]. Hida et al. proposed a dynamic and static topic model (DSTM) for LDA to simultaneously consider the dynamic structures of the temporal topic evolution and the static structures of the topic hierarchy [24]. However, none of these approaches seem to address LDA's intrinsic inability to capture semantical information. Guo et al. attempted to exploit dictionary definitions explicitly from WordNet yielding a better understanding of word semantics [25], but WordNet ontologies are too fine-grained resulting in a topic model of less generalisation power. Similarly, Liu et al. attempted to capture the context of the words in documents by incorporating word embedding and part-of-speech in their representation [14]. Indeed this produced impressive results: their topic model with SVM classifier achieved 70.2% accuracy on the 20Newsgroups dataset. As we shall see, this is one of the best classification results for this dataset, but requires a lot of training data and its performance is still lower than the accuracy of our proposed OLDA model when combined with self-training.

A secondary issue with LDA is that the construction of

$$\begin{pmatrix} d_{11}, d_{12}, \ldots, d_{1k} \\ d_{21}, d_{22}, \ldots, d_{2k} \\ \vdots \\ d_{n1}, d_{n2}, \ldots, d_{nk} \end{pmatrix} = \begin{pmatrix} q_{11}, q_{12}, \ldots, q_{1l} \\ q_{21}, q_{22}, \ldots, q_{2l} \\ \vdots \\ q_{n1}, q_{n2}, \ldots, q_{nl} \end{pmatrix} \times \begin{pmatrix} p_{11}, p_{12}, \ldots, p_{1k} \\ p_{21}, p_{22}, \ldots, p_{2k} \\ \vdots \\ p_{l1}, p_{l2}, \ldots, p_{lk} \end{pmatrix}$$

Δ (document/words) $\quad$ Θ (documents/topics) $\quad$ Φ (topics/words)

Fig. 1. *A typical schematic of LDA matrices*

the topic model in a purely unsupervised manner results in a rather inaccurate topic classification. Several approaches have attempted to create semi-supervised models. Wang et al. suggested that the construction of the topic model could be improved by manually incorporating available expert knowledge (ssLDA, [12]), but this still requires a lot of human intervention. Fu et al. fixed the number of topics across datasets [26], whereas Wu et al. represented documents as concept vectors [27] using heuristic selection rules to select only related keywords rather than the full-text obtained from Wikipedia. Gu et al. combined a supervised Bi-directional Recurrent Neural Network (Bi-RNN), a neural network with Long Short-Term Memory (LSTM), and LDA to capture contextual information and discover latent semantic information in the representation of short documents [28]. Finally, Pavlinek et al. suggested the use of a self-training algorithm within LDA (ST-LDA, [11]). To the best of our knowledge, this was the first approach to use self-training in topic modelling. It is worth noting that the self-training process, although done automatically, can also be very time consuming in LDA. As we shall see, the proposed incorporation of the ontological component within OLDA can significantly reduce this time whilst achieving higher classification accuracy at the same time.

III. Ontology-Driven Approach for Topic Classification

Our classification approach addresses the two issues with pure LDA mentioned in Section II, namely its inability to consider different word meanings, and the amount of human supervision needed to train the model. In order to address the first problem, we introduce an intermediate step to the topic-modelling process using labels representing concepts of an ontology to capture the different meanings of the words. For this reason, our technique can be considered an *ontology-driven* variant of LDA, which we abbreviate to OLDA. In its pure approach, it also requires human supervision, but as for LDA, it also allows the incorporation of a self-training phase. Accordingly, we refer to this self-training variant as ST-OLDA.

OLDA's aim is to generate a documents/topics matrix Θ giving the probability q_{xy} of each document D_x being about a certain topic T_y. The incorporation of the ontological concepts is done through the introduction of an intermediate matrix in the LDA scheme of Fig. 1 resulting in the scheme in Fig. 2. We first pre-process the documents employing standard open source NLP tools (StanfordNLP [29]) for part-of-speech (POS) tagging and extracting the set $\mathscr{W}$ of all words in them. As before, we construct the matrix Δ of binary values, where each cell d_{ij} is given the value 1 if the document D_i ($1 \leq i \leq n$) contains the word W_j ($1 \leq j \leq k$) or 0, otherwise. Using DBpedia [30], we then construct the set of all labels $\mathscr{L} = \{L_1, \ldots, L_m\}$ that are associated with a word $W \in \mathscr{W}$. Analogously, we then construct the matrix Γ of binary values, where each cell s_{ro} is given value 1 if the word W_o ($1 \leq o \leq k$) can be described by the label L_r ($1 \leq r \leq m$), or 0 otherwise (this process is described in more detail in Section III-A). The matrix Σ giving the probabilities r_{ab} of each topic T_a being described by each label l_b is constructed using a *logistic regression technique*. Finally, Θ is computed by a supervised learning method using Δ, Σ and Γ (the computation of Σ and Θ are described in Section III-B). In Section IV we explain how the amount of human supervision needed can be minimised.

A. Generating the Labels/Words Matrix Γ

DBpedia is a crowd-sourced community website providing structured content extracted from the information created in various Wikipedia projects. This structured knowledge is freely available for use and described by a shallow, cross-domain ontology called the *DBpedia Ontology*. The DBpedia Ontology currently consists of 685 concepts described by 2795 different properties. An important property of each concept is its *Type*, which loosely describes the semantic meaning of the concept. We use the type property of the concepts to create the set of labels $\mathscr{L}$ and the matrix Γ which gives the association between words and labels as follows (this process is done programmatically via scripts without human intervention).

For each noun $W \in \mathscr{W}$, we query DBpedia to obtain W's type properties and then construct the set of labels $L(W)$ associated with the word W. Because of the way the type properties are presented in DBpedia, some basic data cleansing is needed: we remove any redundant information in the property; segment words as needed; and aggregate similar terms. For example, the word "computer" has ten different Type properties in DBpedia: *Thing; Device; Artifact100021939; ComputerSystem103085915; Instrumentality103575240; Object100002684; PhysicalEntity100001930; System104377057; Whole100003553 and WikicatComputerSystems*. During cleansing we remove the numerical references from the types; segment words in terms such as "ComputerSystem"; and combine similar words such as "System" and "Systems" into a single label. For our "com-

$$\begin{pmatrix} d_{11}, d_{12}, \ldots, d_{1k} \\ d_{21}, d_{22}, \ldots, d_{2k} \\ \vdots \\ d_{n1}, d_{n2}, \ldots, d_{nk} \end{pmatrix} = \begin{pmatrix} q_{11}, q_{12}, \ldots, q_{1l} \\ q_{21}, q_{22}, \ldots, q_{2l} \\ \vdots \\ q_{n1}, q_{n2}, \ldots, q_{nl} \end{pmatrix} \times \begin{pmatrix} r_{11}, r_{12}, \ldots, r_{1m} \\ r_{21}, r_{22}, \ldots, r_{2m} \\ \vdots \\ r_{l1}, r_{l2}, \ldots, r_{lm} \end{pmatrix} \times \begin{pmatrix} s_{11}, s_{12}, \ldots, s_{1k} \\ s_{21}, s_{22}, \ldots, s_{2k} \\ \vdots \\ s_{m1}, s_{m2}, \ldots, s_{mk} \end{pmatrix}$$

Δ (documents/words) $\quad \Theta$ (documents/topics) $\quad \Sigma$ (topics/labels) $\quad \Gamma$ (labels/words)

Fig. 2. *Ontology-Driven topic model matrices schematic*

puter" example, we would end up with the set of labels $L(computer) = \{Thing, Device, Artifact, Computer\ System, Instrumentality, Object, Physical\ Entity, System, \ldots\}$.

We then set $\mathscr{L} = \bigcup_{W \in \mathscr{W}} L(W)$; assume a fixed ordering of labels $[L_0, L_1, \ldots, L_m]$ (for $L_i \in \mathscr{L}$); and then construct the matrix Γ by setting $s_{ij} = 1$ if $L_j \in L(W_i)$, or 0 otherwise.

B. Generating the Matrices Θ and Σ

The documents/topics matrix Θ and the topics/labels matrix Σ are generated iteratively using the input matrix Γ in a logistic regression model. This is a machine learning technique that tries to capture patterns within data features. The model uses the linear weighted combination of inputs from Γ and generates the predicted probabilities of each label relating to each topic (i.e., the matrix Σ) [31], [32]. A schematic diagram of the model is shown in Fig. 3. For each word $W_i \in \mathscr{W}$, the corresponding column in the labels/words matrix Γ is used as an input data vector $\mathbf{L_i}$ (equation (1) below) to generate an output vector $\mathbf{y_i}$ with the predicted probability of each label being associated with a topic, as described next. A fully connected layer takes the vector $\mathbf{L_i}$ and generates the evidence vector $\mathbf{z_i}$ using (2) and a weight matrix $\mathbf{W}_t$ and bias vector b_t. The initial values $\mathbf{W}_0$ and b_0 are randomly given.

$$\mathbf{L_i}_j = \begin{cases} 1 & \text{if } l_j \in L(W_i) \\ 0 & \text{if } l_j \notin L(W_i) \end{cases} \tag{1}$$

$$\mathbf{z}_i = f(\mathbf{L_i}) = \mathbf{W}_t \mathbf{L_i} + b_t \tag{2}$$

Each element $\hat{r}_a$ in the evidence vector $\mathbf{z_i}$ is then normalised in the softmax layer to finally generate the vector $\mathbf{y_i}$ according to (3) (this means that the values within $\mathbf{y_i}$ add up to 1). Each element r_{ab} in the output vector $\mathbf{y_i}$ is the predicted probability of each label l_b being associated with a topic T_a. This whole process is repeated for all words ($i \in (1, 2, \ldots, k)$), resulting in the matrix Σ_t. Finally, the matrix Θ_t can be computed using the matrix schematic shown in Fig. 2.

$$\mathbf{y_i} = softmax(\mathbf{z}_i) = \frac{e^{\hat{r}_a}}{\sum_{a=1}^{l} e^{\hat{r}_a}} \tag{3}$$

Consequently, the initial matrices Σ_0 and Θ_0 are obtained using random values for the weight matrix $\mathbf{W}_0$ and the bias vector b_0. For each subsequent iteration $t+1$, we then measure the Euclidean distance between the predicted classification Θ_t and the true classification Θ_s (recall Θ_s is manually done). Using the Stochastic Gradient Descent technique we obtain new values for $\mathbf{W}_{t+1}$ and the vector b_{t+1} [33], [34] that minimise this distance. We then calculate Σ_{t+1} and Θ_{t+1} as before using $\mathbf{W}_{t+1}$ and b_{t+1}. This process continues until the distance between the predicated classification Θ_j computed in an iteration j and the true classification Θ_s goes below a desired threshold. The output of this process is the documents/topics matrix Θ, the topics/labels matrix Σ, and the optimised weight matrix $\mathbf{W}$ and bias vector b.

IV. Reducing the Required Amount of Pre-classified Data

Obviously, obtaining a large training dataset is costly, so we would like to minimise the amount of pre-classified data required. As done for LDA in the creation of ST-LDA, this can be done by introducing a self-training stage to enlarge the original amount of manually trained data.

In this section, we consider the use of two self-training approaches for this: the first, presented in Section IV-A, consists of a relatively ad hoc procedure that is quick to perform and produces good but sub-optimal results. As an alternative, we also describe Pavlinek et al.'s self-training procedure [11] in Section IV-B. We will see that this procedure takes nearly twice as long to complete as the ad hoc method, but produces the best results when combined with OLDA. We stress that the introduction of the labelling matrix reduces the time required for training by approximately half independently of the training procedure used.

A. A Simple Self-training Procedure

Instead of using a large amount of pre-classified documents to calculate values for the weight matrix $\mathbf{W}$ and the bias vector b, the idea is to take a very small amount of pre-classified documents D_s and a much larger amount of unclassified documents D_u to output an enlarged classified training matrix Θ_{ss} from the manually provided matrix Θ_s. $\mathbf{W}$ and b are obtained from D_s and Θ_s as described in Section III-B until we reduce the distance between Θ_t and Θ_s to below 0.55 (this choice of value is explained in Section V-B).

In a second phase, we use the values of $\mathbf{W}$ and b thus obtained to automatically train the remaining unclassified data (D_u). Thus, the final training set D_{ss} consists of the manually classified set D_s together with the automatically trained set D_u and is applicable for training purposes as in any other supervised classification method. The resulting topic model Θ_{ss} and Σ_{ss} can then be used to classify the remaining unclassified documents.

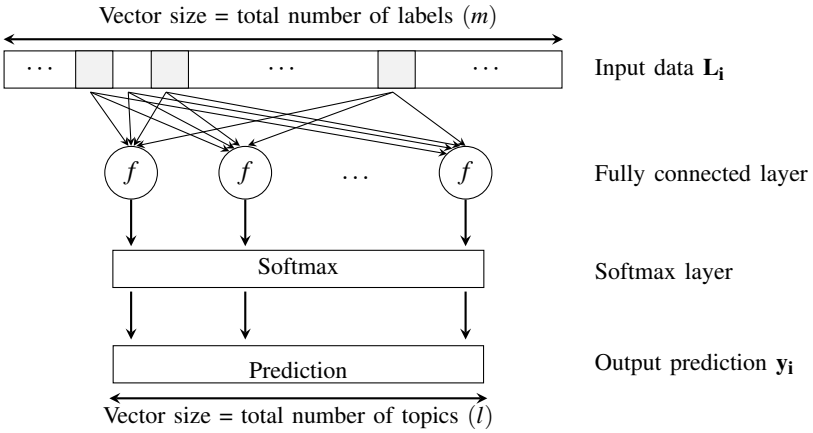

Fig. 3. *Structure of logistic regression model*

B. The Self-training Procedure Proposed by Pavlinek

Pavlinek et al. proposed a more elaborate self-training algorithm also consisting of two phases [11]. As before, the goal of the first phase is to generate a topic model from the smaller amount of manually classified data D_s. However, they employ Gibbs sampling [35] to do this.

In the second phase, unclassified data (from D_u) is iteratively classified using the topic model generated in the first phase and compared using a centroids distance until a predefined threshold is reached. The centroid distance is defined by a semantic similarity measure based on the topic distribution and a cosine similarity measure defined in terms of the centroids for each category [36].

The second phase finishes when for each unclassified document in D_u, the difference between the distances from the two nearest centroids is smaller than the similarity threshold. As a result of this phase, we also end up with an enlarged classified training set D_{ss} consisting of the manually classified set D_s and the automatically classified set D_u. Full details of the whole process can be found in [11].

V. Experimental Analysis

As we mentioned, OLDA can be used with or without a self-training stage. When self-training is employed we use the prefix "ST" and refer to the resulting classification method as ST-OLDA instead. As we suggested two different self-training procedures, the prefix ST is subscripted with H (to indicate the use of the ad hoc training procedure) or P (to indicate the use of Pavlinek et. al.'s). Where the distinction is irrelevant, we avoid the subscript. With all this in mind, we conducted comprehensive benchmarking to evaluate the performance of our method and variants against a number of other semi-supervised methods using four different widely available datasets.

More specifically, we compared the performance of OLDA with the performance of ST-OLDA with the Bag-of-Words (BOW) representation with a Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme [37] and with the Latent Dirichlet Allocation (LDA) [11]. Furthermore, we considered the two self-training techniques (ST_H and ST_P) described in Section IV for LDA, TF-IDF and OLDA, resulting in a total of six variations of the semi-supervised methods. The results of our experiments are given in Tables I and II and discussed in more detail in Section V-B. All experiments were performed on a PC with an i7 processor, a NVIDIA GeForce GPU GTX 970M graphics card, and 16GB RAM.

A. Datasets Used in the Analysis

In our analysis we used the 20 Newsgroups dataset, the Reuters R8 and R52 datasets and the WebKB dataset. For each dataset, we performed some preprocessing to combine word variants and to remove words that we deemed irrelevant. To be precise: (i) all words were converted to lower case; (ii) stop words (such as "etc.", "I'm" and "of") were removed; (iii) words shorter than three characters were also removed; and (iv) plural words were converted into singular using lemmatization tool from StanfordNLP. We now briefly describe how each of these datasets were used.

20 Newsgroups: This dataset comprises a collection of $18,846$ newsgroup documents, partitioned (nearly) evenly across 20 different categories, each corresponding to a different topic. We used the so-called "bydate" version, where duplicates and some headers are removed. In total, $233,745$ words were extracted from the documents. After preprocessing the number of words was reduced to $155,387$ and hence our documents/words matrix Δ is a $18,846 \times 155,387$ binary matrix. We then extracted $13,820$ labels in DBpedia associated with these words, yielding a labels/words matrix Γ of size $13,820 \times 155,387$ for the ST-OLDA methods. For each round of experiments for ST TF-IDF, ST-LDA and ST-OLDA, we used 5% of the data for the first phase of training, 45% for the second semi-supervised phase, and used the remaining 50% for testing.

Reuters R8 and **Reuters R52**:These two datasets are derived from the Reuters-21578 dataset and are single labelled with a ModApte split, which means that each topic category contains at least two documents and hence at least one can be used for training and one for testing. Reuters R8 contains

TABLE I
CLASSIFICATION ACCURACY RESULTS

Dataset	Semi-supervised ST_H			Semi-supervised ST_P		
	TF-IDF	LDA	OLDA	TF-IDF	LDA	OLDA
20 Newsgroups	56.21%	61.33%	**72.12%**	60.25%	68.51%	**78.01%**
Reuters R8	20.11%	60.54%	**77.32%**	23.66%	75.71%	**83.11%**
Reuters R52	22.05%	45.88%	**56.14%**	25.87%	53.24%	**64.25%**
WebKB	63.31%	68.09%	**74.90%**	67.13%	72.38%	**81.89%**

7674 documents divided into 8 categories. From the words extracted, 7808 were left after preprocessing, resulting in a 7674×7808 binary documents/words matrix Δ. We then found 5315 labels associated with these words in DBpedia, yielding a 5315×7808 labels/words matrix Γ. Reuters R52 consists of 9100 documents divided into 52 categories. Preprocessing of the words extracted resulted in 8937 words yielding a 9100×8937 binary documents/words matrix Δ. As before, we extracted 6471 associated labels from DBpedia, yielding a 6471×8937 labels/words matrix Γ. With these datasets, we employed an approximate 70/30 ratio for training/testing as normally employed elsewhere. For ST TF-IDF, ST-LDA and ST-OLDA, we used 7% of the data for the first phase of training, 63% for the second phase, leaving the remaining 30% for testing.

WebKB: This dataset comprises a collection of websites from computer science departments, whose pages are divided into seven categories: student, faculty, staff, course, project, department and other. Our experiments used a variant of the dataset covering 4199 documents from the first four previous categories. After preprocessing, we were left with a 4199×7719 binary documents/words matrix Δ. We found 5109 associated labels in DBpedia, yielding a 5109×7719 labels/words matrix Γ. For ST TF-IDF, ST-LDA and ST-OLDA, we used 6% of the data for the first phase of training, 60% for the second phase, and the remaining 33% for testing.

B. Experimental Results

We conducted two rounds of experiments with each of the four datasets. In each round of the semi-supervised experiments we performed 10 repetitions in the training and selected the data for training using stratified random sampling for each topic category, so that each topic had equal representation in the training set.

Table I summarises the classification accuracy results of TF-IDF, LDA and OLDA when using either of the two self-training procedures ST_H and ST_P. Table II summarises the topic model's construction times for each techniques for the 20Newsgroup dataset.

In what follows, we discuss the results using each self-training procedure in more detail.

1) Self-Training Using the Simplified Approach (ST_H): As we mentioned in Section IV-A, the training procedure stops when the distance between the predicted and actual classification drops below a certain threshold. In our experiments, this distance drops dramatically in the first $2,500,000$ iterations, decreasing further but at a reduced rate in later iterations. The distance remained fairly stable after $20,000,000$ iterations dropping to values close to 0.54. For that reason, we stop iterating when the distance goes below 0.55.

As shown in Table I, TF-IDF performed worst of all in all datasets, with OLDA also outperforming LDA by quite a considerable margin (e.g., 77.32% against 60.54% in the Reuters R8 dataset). As shown in Table II, the construction of the topic model for the 20 Newsgroups dataset using the training procedure ST_H for OLDA took about two days to complete while it took five days for LDA. This is 40% of the time.

2) Self-Training Using Pavlinek et al.'s Approach (ST_P): OLDA's construction of the topic model for the 20Newsgroup dataset using the training procedure ST_P took about five days, whilst LDA's took ten days. That is, OLDA's construction took around half the time.

In terms of accuracy, the training procedure ST_P performed better in all techniques and datasets. TF-IDF performed worst in all datasets albeit it was better when using the training procedure ST_P than when using ST_H. The best combination was ST_P and OLDA, which outperformed ST_P and LDA by quite a considerable margin (64.25% against 53.24% in the Reuters R52 dataset).

TABLE II
TIME TO CONSTRUCT THE 20 NEWSGROUPS TOPIC MODEL

Technique	Construction time (days)	
Semi-supervised ST_H	TF-IDF	2
	LDA	5
	OLDA	2
Semi-supervised ST_P	TF-IDF	6
	LDA	10
	OLDA	5

So we can conclude that the self-training procedure ST_P is superior to the simple training procedure ST_H although its topic model takes roughly twice as long to construct. We can also conclude that the introduction of the intermediate ontology concepts to the topic model helps to reduce the amount of time required to train the model (independently of the self-training procedure employed).

So our overall conclusion is that the introduction of the concept matrix into the topic model not only increases the accuracy of the classification across all datasets but also helps to reduce the training time by up to 60%.

VI. Conclusions and Future Work

Conventional data-driven approaches to topic modelling of natural language texts, such as Term Frequency - Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA), come with two important limitations. Firstly, these approaches do not use the semantical meanings of the words, ignoring the fact that individual words may have multiple meanings and that different words may have the same meaning. This limits the ability of the method to perform the modelling independently of the particular set of words describing the topics. Secondly, they require a significant amount of classified training data for supervised machine learning. Generating this training data is expensive and time-consuming as it relies on humans to collect, read and manually classify the data in a consistent manner.

In this paper we propose a novel approach based on LDA that uses ontological information obtained from DBpedia about the semantical meaning of the words, allowing topics to be represented more faithfully and independently to the particular set of words used to describe them. This approach, that we called Ontology-Driven Latent Dirichlet Allocation (OLDA), can be combined with a self-training phase to produce a semi-supervised method (ST-OLDA), which requires only a small amount of pre-classified training data. The idea is to generate the topic model using the restricted amount of manually classified data – typically, only 10% of the training data, and then use the remaining 90% of the training data to automatically train the model. The resulting model is then used to classify the remaining testing data.

Our experiments, using the four datasets "20 Newsgroups", "Reuters R8", "Reuters R52" and "WebKB", show that the addition of the semantical component into LDA significantly increases the accuracy of the classification. In addition, when used with self-training, this allows the reduction of the amount of trained data needed and significantly increases the performance of the classification over ST-LDA and ST TF-IDF, while reducing the time required for training.

Our main conclusions can be summarised as follows:

1) The inclusion of the ontological component reduces the self-training time by nearly half using two distinct self-training procedures. In particular, it reduces the time needed for training using the self-training procedure proposed by [11] by nearly half in the 20 Newsgroups dataset.

2) The inclusion of the ontological component also increases the accuracy of the classification regardless of the self-training method employed by between 6 and 17 percentual points (depending on the training method and dataset).

3) The self-training procedure proposed by [11] produces better accuracy results than an Ad Hoc procedure suggested in this paper, for both LDA and OLDA, independently of the dataset, although it takes twice as long to train. When combined with OLDA it provides the best accuracy results in all datasets, significantly outperfoming ST-LDA.

These results are very encouraging and we think that there is scope for further improvement of the classification accuracy through the use of specialised ontologies and/or a more fine-tuned selection of labels. Furthermore, incorporating the relationships between words and ontological concepts into the topic model is also an interesting future direction. These are left as future work.

References

[1] D. Lazer, D. Brewer, N. Christakis, J. Fowler, and G. King, "Life in the network: the coming age of computational social," *Science*, vol. 323, no. 5915, pp. 721–723, 2009.

[2] Q. Li, S. Shah, X. Liu, A. Nourbakhsh, and R. Fang, "Tweet topic classification using distributed language representations," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2016, pp. 81–88.

[3] C.-I. Hsu and C. Chiu, "A hybrid latent dirichlet allocation approach for topic classification," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2017, pp. 312–315.

[4] H. Wang and K. Wong, "Recommendation-assisted personal web," in *2013 IEEE Ninth World Congress on Services*. IEEE, 2013, pp. 136–140.

[5] Q. Lu, J. G. Conrad, K. Al-Kofahi, and W. Keenan, "Legal document clustering with built-in topic segmentation," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 383–392.

[6] S. Burkhardt and S. Kramer, "Online multi-label dependency topic models for text classification," *Machine Learning*, vol. 107, no. 5, pp. 859–886, 2018.

[7] M. Girolami and A. Kabán, "On an equivalence between plsi and lda," in *SIGIR*, vol. 3, 2003, pp. 433–434.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[9] Y. Ko and J. Seo, "Text classification from unlabeled documents with bootstrapping and feature projection techniques," *Information Processing & Management*, vol. 45, no. 1, pp. 70–83, 2009.

[10] U. Ocepek, J. Rugelj, and Z. Bosnić, "Improving matrix factorization recommendations for examples in cold start," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6784–6794, 2015.

[11] M. Pavlinek and V. Podgorelec, "Text classification method based on self-training and LDA topic models," *Expert Systems with Applications*, vol. 80, pp. 83–93, 2017.

[12] D. Wang, M. Thint, and A. Al-Rubaie, "Semi-supervised latent dirichlet allocation and its application for document classification," in *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, 2012, pp. 306–310.

[13] J. C. Campbell, A. Hindle, and E. Stroulia, "Latent dirichlet allocation: extracting topics from software engineering data," in *The art and science of analyzing software data*. Elsevier, 2015, pp. 139–159.

[14] W. Liu, P. Liu, Y. Yang, J. Yi, and Z. Zhu, "A¡ word, part of speech¿ embedding model for text classification," *Expert Systems*, p. e12460, 2019.

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[16] W. Sriurai, "Improving text categorization by using a topic model," *Advanced Computing*, vol. 2, no. 6, p. 21, 2011.

[17] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[19] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[20] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 248–256.
[21] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia, "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms," in *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013, pp. 522–531.
[22] C.-I. Hsu and C. Chiu, "A hybrid latent dirichlet allocation approach for topic classification," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2017, pp. 312–315.
[23] K. Krasnashchok and S. Jouili, "Improving topic quality by promoting named entities in topic modeling," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2018, pp. 247–253.
[24] R. Hida, N. Takeishi, T. Yairi, and K. Hori, "Dynamic and static topic model for analyzing time-series document collections," *arXiv preprint arXiv:1805.02203*, 2018.
[25] W. Guo and M. Diab, "Semantic topic models: Combining word distributional statistics and dictionary definitions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 552–561.
[26] Y. Fu, M. Yan, X. Zhang, L. Xu, D. Yang, and J. D. Kymer, "Automated classification of software change messages by semi-supervised latent dirichlet allocation," *Information and Software Technology*, vol. 57, pp. 369–377, 2015.
[27] Z. Wu, H. Zhu, G. Li, Z. Cui, H. Huang, J. Li, E. Chen, and G. Xu, "An efficient wikipedia semantic matching approach to text document classification," *Information Sciences*, vol. 393, pp. 15–28, 2017.
[28] Y. Gu, M. Gu, Y. Long, G. Xu, Z. Yang, J. Zhou, and W. Qu, "An enhanced short text categorization model with deep abundant representation," *World Wide Web*, vol. 21, no. 6, pp. 1705–1719, 2018.
[29] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.
[30] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.
[31] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1-2, pp. 167–179, 1967.
[32] S. Menard, *Applied logistic regression analysis*. Sage, 2002, vol. 106.
[33] J. Kiefer, J. Wolfowitz *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.
[34] L. Bottou, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, no. 9, p. 142, 1998.
[35] G. Casella and E. I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.
[36] E.-H. S. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," in *European conference on principles of data mining and knowledge discovery*. Springer, 2000, pp. 424–431.
[37] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

Qi Hao received the B.Sc. degree in information engineering from Beijing Institute of Technology, Beijing, China. in 2014, and the M.Sc. degree in analogue and digital integrated circuit design from Imperial College London, London, U.K., in 2015. She was also a exchange student in Department of Electrical Electronics Engineering, University of California, Berkeley from August, 2013 to May 2014. She is currently pursuing the Ph.D. degree at the Department of Informatics, King's College London, with a focus on Natural Language Processing and machine learning for topic classification.

Dr. Jeroen Keppens is a Senior Lecturer at the Department of Informatics, King's College London. His work focuses on evidential reasoning, reasoning under uncertainty and explanation of inference engines of decision support systems. Dr. Keppens has a particular interest in the integration of Bayesian, argumentation and narrative approaches to analysing the value evidence in Law. His work has also been applied to ecological modelling, mental health and statistical model selection.

Dr. Odinaldo Rodrigues is a Senior Lecturer at the Department of Informatics, King's College London. He is an expert in the areas of knowledge representation and reasoning; argumentation theory; and computation of argumentation semantics. Dr. Rodrigues has worked in several research projects integrating logic and complex algorithms and his work focuses on techniques used in the formalisation of the common-sense reasoning, such as belief revision in classical and non-classical logics, abduction, argumentation and also applications of AI to software engineering and social choice theory. Dr. Rodrigues is executive editor of the Journal of Logic and Computation and the Logic Journal of the IGPL (Oxford Academic Journals).