On Horizontal and Vertical Separation in Hierarchical Text Classification

Mostafa Dehghani[1] Hosein Azarbonyad[2] Jaap Kamps[1] Maarten Marx[2]
[1]Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands
[2]Informatics Institute, University of Amsterdam, The Netherlands
{dehghani,h.azarbonyad,kamps,maartenmarx}@uva.nl

ABSTRACT

Hierarchy is a common and effective way of organizing data and representing their relationships at different levels of abstraction. However, hierarchical data dependencies cause difficulties in the estimation of "separable" models that can distinguish between the entities in the hierarchy. Extracting separable models of hierarchical entities requires us to take their relative position into account and to consider the different types of dependencies in the hierarchy. In this paper, we present an investigation of the effect of separability in text-based entity classification and argue that in hierarchical classification, a separation property should be established between entities not only in the same layer, but also in different layers.

Our main findings are the followings. First, we analyse the importance of separability on the data representation in the task of classification and based on that, we introduce a "Strong Separation Principle" for optimizing expected effectiveness of classifiers decision based on separation property. Second, we present Hierarchical Significant Words Language Models (HSWLM) which capture all, and only, the essential features of hierarchical entities according to their relative position in the hierarchy resulting in horizontally and vertically separable models. Third, we validate our claims on real world data and demonstrate that how HSWLM improves the accuracy of classification and how it provides transferable models over time. Although discussions in this paper focus on the classification problem, the models are applicable to any information access tasks on data that has, or can be mapped to, a hierarchical structure.

Keywords

Separation, Hierarchical Significant Words Language Models, Hierarchical Text Classification

1. INTRODUCTION

Hierarchy is an effective and common way of representing information and many real-world textual data can be organized in this way. Organizing data in a hierarchical structure is valuable since it determines relationships in the data at different levels of resolution and picks out different categories relevant to each of the different layers of memberships. In a hierarchical structure, a node at any layer could be an indicator of a document, a person, an organization, a category, an ideology, and so on, which we refer to them as "hierarchical entities". Taking advantage of the structure in the hierarchy requires a proper way for modeling and representing entities, taking their relation in the hierarchy into consideration.

There are two types of dependencies in the hierarchies: i) *Horizontal dependency*, which refers to the relations of entities in the same layer. A simple example would be the dependency between siblings which have some commonalities in terms of being descendants of the same entity. ii) *Vertical dependency*, which addresses the relations between ancestors and descendants in the hierarchy. For example the relation between root and other entities. Due to the existence of two-dimensional dependencies between entities in the hierarchy, modeling them regardless of their relationships might result in overlapping models that are not capable of making different entities distinguishable. Overlap in the models is harmful because when the data representations are not well-separated, classification and retrieval systems are less likely to work well [22]. Thus, *two-dimensional separability*, i.e. *horizontal and vertical separability*, is one of the key requirements of hierarchical classification.

As a concrete example, consider a simple hierarchy of a multi-party parliament as shown in Figure 1, which determines different categories relevant to the different layers of membership in the parliament. We can classify these entities based on text, in particular the transcripts of all speeches in parliament as recorded in the parliamentary proceedings. That is, we can characterize an individual member of parliament by her speeches, a political party by their member's speeches, the opposition by the speeches of members of opposition parties, etc. However, in this way, all classifiers are based on speeches of (set of) individual members, making it important to take relations between different layers of the hierarchy explicitly taken into account. That is, in order to represent a party in this hierarchy, a proper model would show common characteristics of its members—not members of other parties (*horizontal* separation), and capture the party's generic characteristics—not unique aspects of the current members captured in the individual member's layer or aspects of whether the party is in government or opposition captured in the status layer (*vertical* separation).

The concept of separability is of crucial importance in information retrieval, especially when the task is not just ranking of items based on their probability of being relevant, but also making a boolean decision on whether or not an item is relevant, like in information filtering. Regarding this concern, Lewis [23] has presented the Probability Threshold Principle (PTP), as a stronger version of the Probability Ranking Principle [27], for binary classification, which discusses optimizing a threshold for separating items regarding their probability of class membership. PTP is a principle based on the separability in the score space. In this paper, we discuss separability in the data representation and define a *Strong Separation Principle*

ICTIR '16, September 12 - 16, 2016, Newark, DE, USA

ISBN 978-1-4503-4497-5/16/09...$15.00

DOI: http://dx.doi.org/10.1145/2970398.2970408

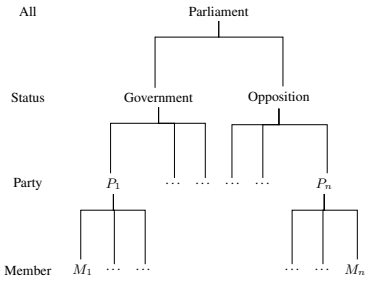
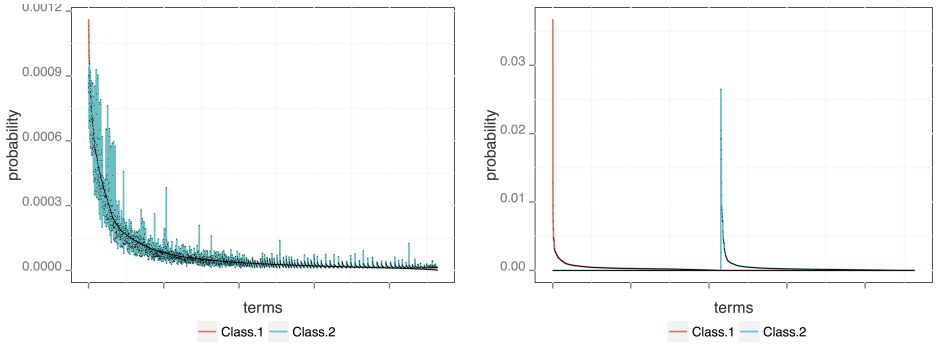


Figure 1: Hierarchical relations in the parliament



(a) A non-separable representation of data

(b) A well-separable representation of data

Figure 2: Probability distribution over terms for data in two different classes, (entities in the statues layer of the parliament), sorted based on the term weights in one of the classes.

as the counterpart of PTP in the feature space.

Separation in the data or feature space is a favorable property that not only helps to improve for ranking or classification algorithms, but also brings out characteristic features for human inspection. Figures 2a and 2b illustrate two different ways of modeling two entities in the status layer of the parliamentary hierarchy, i.e., government and opposition. Each model is a probability distribution over terms (language model) based on the speeches given by all the members in the corresponding status. In each figure, we sort the terms based on their weights in one of the models, and plot the other in the same order. As can be seen, although distributions over terms in Figure 2a for two classes are different, they do not suggest highly separable representations for classes. However, estimated language models in Figure 2b provide highly separable distributions over terms for two classes, identifying the characteristic terms that uniquely represent each class, and can be directly interpreted. Moreover, the language models in Figure 2b select a small set of characteristic features, making it easy to learn effective classifiers for classes of interest.

The main aim of this paper is *to understand and validate the effect of the separation property on hierarchical classification and discuss how to provide horizontally and vertically separable language models for text-based hierarchical entities.* We break this down into three concrete research questions:

RQ1 *What makes separability a desirable property for classifiers?*

We demonstrate that based on the ranking and classification principles, *separation property* in the data representation theoretically follows separation in the scores and consequently improves the accuracy of classifiers' decisions. We state this as the "Strong Separation Principle" for optimizing expected effectiveness of classifiers. Furthermore, we define two-dimensional separation in the hierarchical data and discuss its necessity for hierarchical classification

RQ2 *How can we estimate horizontally and vertically separable language models for the hierarchical entities?*

We show that to estimate horizontally and vertically separable language models, they should capture *all*, and *only*, the essential terms of the entities taking their positions in the hierarchy into consideration. Based on this, extending [11], we introduce Hierarchical Significant Words Language Models (HSWLM) and evaluate them on the real-world data to demonstrate that they provide models for hierarchical entities that possess both horizontal and vertical separability.

RQ3 *How separability improves transferability?*

We investigate the effectiveness of language models of hierarchical entities possessing two-dimensional separation across time and show that separability makes the models capture essential characteristics of a class, which consequently improves transferability over time.

The rest of the paper is structured as follows. Next, in Section 2, we discuss some related work. In Section 3, we argue how separability theoretically improves the accuracy of classification results and discuss horizontal and vertical separation in hierarchical structure. Then, we discuss how to estimate HSWLM as two-dimensionally separable models for hierarchical entities in Section 4. In Section 5, we analyse separability of HSWLM and provide some experiments to assess the transferability of models using HSWLM. Finally, Section 6 concludes the paper and suggests some extensions to this research as the future work.

2. RELATED WORK

This section discusses briefly the separation property in the related domains and review principles in information retrieval and text classification, which are associated with the concept of separability. In addition, some research on classification and topic modeling of hierarchical texts are discussed.

Separability is a property which makes the data representation sufficient to distinguish instances and consequently enables autonomous systems to easily interpreter the data [22]. For instance in the classification task, classifiers learn more accurate data boundaries when they are provided with separable representations of data from different classes [23]. The importance of separability in classifiers has led to the fact that making data separable becomes part of classification. As the most familiar instances, SVM by adding extra dimensions implicitly transform the data into a new space where they are linearly separable [6].

Separation is also a pivoting concept in the information retrieval. Separating relevant from non-relevant documents is a fundamental issue in this domain [21, 27, 28]. In IR, separation plays more important role when instead of giving a rank list, a decision should be made about relevancy of documents, for example in the information filtering task [22]. As another instance, in the task of relevance feedback, there are some efforts on estimating a distinctive model for relevant documents so that it reflects not only their similarity, but also their difference from whole collection, i.e., what makes them stand out or separated [17, 32, 40].

In this paper, we address the separation property in the textual data that are organizable in a hierarchical structure. In a hierarchy, due to the existence of dependencies between entities, estimating separated models is a complex task. There is a range of work on the problem of hierarchical text classification [29, 33], which tried to model hierarchical text-based entities. McCallum et al. [24] proposed a method for modeling an entity in the hierarchy which tackles the

problem of data sparseness in lower layer entities. They used a shrinkage estimator to smooth the model of each leaf entity with the model of its ancestors to make the models more reliable. There is also similar research on XML data processing, as hierarchically structured data, which try to incorporate evidence from other layers as the context through mixing each element language models by its parent's models [25, 30].

Recently, Song and Roth [31] tackled the problem of representing hierarchical entities with a lack of training data for the task of hierarchical classification. In their work, given a collection of instances and a set of hierarchical labels, they tried to embed all entities in a semantic space, then they construct a semantic representation for them to be able to compute meaningful semantic similarity between them.

Zhou et al. [41] proposed a method that directly tackles the difficulty of modeling similar entities at lower levels of the hierarchy. They used regularization so that the model of lower level entities have the same general properties as their ancestors, in addition to some more specific properties. Although these methods tried to model hierarchical texts, but their concerns were not making the models separable. Instead, they mostly addressed the problem of *training data sparseness* [16, 24, 31] or presenting techniques for *handling large scale data* [15, 16, 26, 36].

In terms of modeling hierarchical entities, Kim et al. [20] used Hierarchical Dirichlet Process [HDP, 34] to construct models for entities in the hierarchies using their own models as well as the models of their ancestors. Also, Zavitsanos et al. [39] used HDP to construct the model of entities in a hierarchy employing the models of its descendants. This research tries to bring out precise topic models using structure of the hierarchy, but they do not aim to estimate separable models.

As we will discuss in Section 5, our proposed approach can be employed as a feature selection method for text classification. Prior research on feature selection for textual information [1, 14] tried to improve classification accuracy or computational efficiency, while our method aims to provide a separable representation of data that helps train a transferable model. Apart from considering the hierarchical structure, our goals also differ from prior research on transferability of models. For instance, research on constructing dynamic models for data streams [4, 37] first discovered the topics from data and then tried to efficiently update the models as data changes over the time, while our method aims to identify tiny precise models that are more robust and remain valid over time. Research on domain adaptation [7, 35] also tried to tackle the problem of missing features when very different vocabulary are used in test and training data. This differs from our approach considering the hierarchical relations, as we aim to estimate separable models that are robust against changes in the structure of entities relations, rather than changes in the corpus vocabulary.

3. SEPARABILITY IN THE HIERARCHIES

In this section, we address our first research question: "What makes separability a desirable property for classifiers?"

In addition to the investigation of the separation property in general foundational property of classification and defining a *Strong Separation Principle*, we discuss a two-dimensional separation property of hierarchical classification.

3.1 Separation Property

Separability is a highly desirable property for constructing and operating autonomous information systems [23], and especially classifiers. Here, we present a step by step argument which shows that based on the classification principles, having better separability in the feature space leads to better accuracy in the classification results.

Based on the *Probability Ranking Principle (PRP)* presented by Robertson [27], Lewis [23] has formulated a variant of PRP for binary classification:

> *For a given set of items presented to a binary classification system, there exists a classification of the items such that the probability of class membership for all items assigned to the class is greater than or equal to the probability of class membership for all items not assigned to the class, and the classification has optimal expected effectiveness.*

Since in many applications, autonomous systems need to decide how to classify an individual item in the absence of entire items set, Lewis has extended the PRP to the *Probability Threshold Principle (PTP)*:

> *For a given effectiveness measure, there exists a threshold p, $0 < p < 1$, such that for any set of items, if all and only those items with probability of class membership greater than p are assigned to the class, the expected effectiveness of the classification will be the best possible for that set of items.*

PTP in fact discusses optimizing the effectiveness of classifiers by making items separable regarding their probability of class membership, which is a discussion on "separability in the score space". Based on PTP, optimizing a threshold for separating items is a theoretically trivial task, however, there are practical difficulties. The main difficulty refers to the fact that retrieval models are not necessarily capable of measuring actual probabilities of relevance for documents [2], so they do not guarantee to generate a set of scores from which the optimum cutoff can be inferred. In this regard, a great deal of work has been done on analysing the score distribution over relevant and non-relevant documents to utilize this information for finding the appropriate threshold between relevant and non-relevant documents [2, 3, 19]. It is a clear fact that the more the score distributions of relevant and non-relevant documents are separable, the easier it is to determine the optimum threshold. So, obtaining the *separation property* in the scores distributions of relevant and non-relevant documents is one of the key focus areas for retrieval models.

There are two ways to obtain separability in the scores distributions. We could address the complex underlying process of score generation and investigate ranking functions that yield a separable score distribution, as in the score distributional approaches [e.g. 2]. Alternatively, we can investigate ways to provide existing scoring functions with a highly separable representation of the data. That is, the "term distribution" directly provides information about the "probability of relevance" [8] and if there are separable distributions over $terms$ of relevant and non-relevant documents, a scoring function satisfying PRP will generate scores that separate the classes of relevant and non-relevant documents. Thus, a *separation property* on feature distribution for representing the data is a favorable property, which follows better accuracy of classifiers' decisions.

This paper is a first investigation in the role of separation in the term or feature spaces, in which we introduce a formal definition for separability and formulate a principle on the effectiveness of classification based on separation property and leave a more formal treatment to future work. As a formal and general definition, we can refer to the model separability as follows:

DEFINITION 1. *The model of an entity is epistemologically "separable" if, and only if, it has unique, non-overlapping features that distinguish it from other models.*

We argued that how separability in feature space leads to the separability in score space. Based on this and the given definition of the separability, we present *Strong Separation Principle (SSP)*, which is a counterpart of the PTP [23] in the feature space:

> *For a given set of items presented to a classification system, for each class there exists at least one feature δ in the representation of items, and a threshold τ, such that for any set of items, if all and only those items with $\delta > \tau$ are assigned to the class, the classification will have the optimal possible performance for that set of items in terms of a given effectiveness measure.*

SSP in general is a stronger version of PTP. In strict binary classification, if you have PTP, which holds on the whole feature space, SSP will be satisfied, however in the multi-class case, SSP is stronger and it implies PTP, but not the other way around. Based on PTP, there is always an underlying probabilistic scoring function on the set of *whole* features, which generates membership probabilities as the scores of items and these scores make items separable with regards to a threshold. So, the scoring function can be deemed as a mapping function which maps items to a new feature space in which the score of each item is a single feature representation of that item (membership probabilities (or scores) in PTP would be equivalent to δ in SSP). Thus, when the SSP holds, the PTP and PRP will also hold. One could envision a stronger version of the SSP in which "all" the features in the representations need to be non-overlapping, but the SSP is sufficient for optimizing the effectiveness of the classifier. The separation principle can be formally extended to hierarchical classification in a straightforward way. In the rest of this section, we will discuss the separation property in the hierarchical classification and explain how to estimate separable representations with the aim of satisfying SSP in order to improve the classification effectiveness.

3.2 Horizontal and Vertical Separability

In hierarchical text classification, there are two types of boundaries existing in the data, horizontal boundaries, and vertical boundaries. Hence, a separation property should be established in two dimensions. It means, not only separation between entities' representation in one layer is required, but also separation between the distribution of terms in different layers is needed.

Separation between entities in the same layer is a related concept to the fundamental goal of all classifiers on the data with flat structure, which is making the data in different classes distinguishable [29]. However, separation between entities in different layers is a concept related to difference of abstraction level and modeling data in different layers in a separable way can help the scoring procedures to figure out the meaning behind the layers and make their decisions less affected by the concepts of other unrelated layers, thus leading to conceptually cleaner and theoretically more accurate models.

Based on Definition 1, we formally define horizontal and vertical separability in the hierarchy as follows:

DEFINITION 2. *The model of an entity in the hierarchy is "horizontally separable" if, and only if, it is* separable *compared to other entities in the same layer, with the same abstraction level.*

DEFINITION 3. *The model of an entity in the hierarchy is "vertically separable" if, and only if, it is* separable *compared to other entities in the other layers, with different abstraction levels.*

To formalize these concepts, consider we have a simple three layers hierarchy of text documents with "IsA" relations, where the individual documents take place in the lowest layer, and each node in the middle layer determines a category, representing a group of documents, i.e. its children, and the super node on the top of the hierarchy deemed to represent all the documents in all the groups in the hierarchy. There is a key point in this hierarchy to which we will refer for estimating models for the hierarchical entities: "each node in the hierarchy is a general representation of its descendants".

First assume that the goal is to estimate a language model representing category c, as one of the entities in the middle layer of the hierarchy, and we need the estimated model possessing *horizontal separability*. To estimate a horizontally separable model of a category, which represents the category in a way that it is distinguishable from other categories in the middle layer, the key strategy is to eliminate terms that are common across all the categories (overlapping features) and preserve only the discriminating ones.

To do so, we consider there is a general model that captures all the *common* terms of all the categories in the middle layer, θ_c^g. Also we assume that the standard language model of c, i.e the model estimated from concatenation of all the documents in c using MLE, θ_c, is drawn from the mixture of the *latent horizontally separable model*, θ_c^{hs}, and general model that represents shared terms of all categories, i.e. θ_c^g:

$$p(t|\theta_c) = \lambda p(t|\theta_c^{hs}) + (1-\lambda)p(t|\theta_c^g), \tag{1}$$

where λ is the mixture coefficient. Regarding the meaning of the relations between nodes in the hierarchy, top node in the hierarchy is supposed to be a general representation of all categories. On the other hand θ_c^g supposed to be a model capturing general features of all the categories in the middle layer. Thus, we can approximate θ_c^g with the estimated model of the top node in the hierarchy, θ_{all}:

$$p(t|\theta_c) \approx \lambda p(t|\theta_c^{hs}) + (1-\lambda)p(t|\theta_{all}). \tag{2}$$

We estimate θ_{all} using MLE as follows:

$$p(t|\theta_{all}) = \frac{tf(t,all)}{\sum_{t'} tf(t',all)} = \frac{\sum_{c\in all}\sum_{d\in c} tf(t,d)}{\sum_{c\in all}\sum_{d\in c}\sum_{t'\in d} tf(t',d)}, \tag{3}$$

where $tf(t,d)$ indicates the frequency of term t in document d and θ_{all} is in fact collection language model.

Now, the goal is to extract θ_c^{hs}. With regard to the generative models, when a term t is generated using the mixture model in Equation 2, first a model is chosen based on λ and then the term is sampled using the chosen model. The log-likelihood function for generating the whole category c is:

$$\log p(t|\theta_c^{hs}) = \sum_{t\in c} tf(t,c)\log\big(\lambda p(t|\theta_c^{hs}) + (1-\lambda)p(t|\theta_{all})\big), \tag{4}$$

where $tf(t,c)$ is the frequency of occurrence of term t in category c. With the goal of maximizing this likelihood function, the maximum likelihood estimation of $p(c|\theta_c^{hs})$ can be computed using the Expectation-Maximization (EM) algorithm by iterating over the following steps:

E-step:

$$e_t = tf(t|c).\frac{\lambda p(t|\theta_c^{hs})}{\lambda p(x|\theta_c^{hs}) + (1-\lambda)p(x|\theta_{all})}, \tag{5}$$

M-step:

$$p(x|\theta_c^{hs}) = \frac{e_t}{\sum_{t'\in\mathcal{V}} e_t'}, \text{ i.e. normalizing the model,} \tag{6}$$

where $\mathcal{V}$ is the set of all terms with non-zero probability in θ_c. In Equation 5, θ_c is the maximum likelihood estimation of category

Algorithm Modified Model Parsimonization

procedure PARSIMONIZE(e,B)
  for all term t in the vocabulary **do**
    $P(t|\theta_B) \xleftarrow{normalized} \sum_{b_i \in B} \left(P(t|\theta_{b_i}) \prod_{\substack{b_j \in B \\ j \neq i}} (1 - P(t|\theta_{b_j})) \right)$
    repeat
      E-Step: $P[t \in \mathcal{V}] \leftarrow P(t|\theta_e) . \frac{\alpha P(t|\tilde{\theta}_e)}{\alpha P(t|\tilde{\theta}_e) + (1-\alpha) P(t|\theta_B)}$
      M-Step: $P(t|\tilde{\theta}_e) \leftarrow \frac{P[t \in \mathcal{V}]}{\sum_{t' \in \mathcal{V}} P[t' \in \mathcal{V}]}$
    until $\tilde{\theta}_t$ becomes stable
  end for
end procedure

Figure 3: Pseudo-code for procedure of modified model parsimonization.

Algorithm Estimating Hierarchical Significant Words Language Models

procedure ESTIMATEHSWLMS
  Initialization:
  for all entity e in the hierarchy **do**
    $\theta_e \leftarrow$ standard estimation for e using MLE
  end for
  repeat
    SPECIFICATION
    GENERALIZATION
  until models do not change significantly anymore
end procedure

Figure 4: Pseudo-code for the overall procedure of estimating HSWLM.

c: $p(t|\theta_c) = \sum_{d \in c} c(t,d) / \sum_{d \in c} \sum_{t' \in d} c(t',d)$ and θ_c^{hs} represents the horizontally separable model, which in the first iteration it is initialized by the maximum likelihood estimation, similar to θ_c.

Considering the above process, a horizontally separable model is a model which is **specified** by taking out general features that have high probability in "all" categories, or lets say collection language model, which is similar to the concept of the parsimonious language model, introduced by Hiemstra et al. [17].

Now assume that we want to extract a language model possessing *vertical separability* for the category c, i.e. a model that makes this category distinguishable from entities both in the lower layer (each individual document) and the top layer (collection of all documents). In the procedure of making the model horizontally separable, we argued that we can reduce the problem to removing terms representing the top node, which results in a model that is separable from the top node in the upper layer. This means that we are already half-way towards making a model vertically separable, thus, the model only requires to be separable from its descendant entities in the lower layer. Back to the meaning of the "IsA" relations in the hierarchy, the model of each node is a general representation of all its descendants. So making the model of a category separable from its descendant documents means to remove terms that describe each individual documents, but not all of them. We call these terms, document specific terms. For each category c, we assume there is a model, θ_d^s, that captures document specific terms, i.e. terms from documents in that category that are good indicators for individual documents but not supported by all of them. Also we assume that the standard language model of c, θ_c, is drawn from the mixture of the *latent vertically separable model*, θ_c^{vs}, and θ_d^s:

$$p(t|\theta_c) = \lambda p(t|\theta_c^{vs}) + (1-\lambda) p(t|\theta_d^s) \quad (7)$$

where λ is the mixing coefficient. We estimate θ_d^s using the following equation:

$$p(t|\theta_d^s) \xleftarrow{normalized} \sum_{d_i \in c} \left(p(t|\theta_{d_i}) \prod_{\substack{d_j \in c \\ j \neq i}} (1 - p(t|\theta_{d_j})) \right), \quad (8)$$

where $p(t|\theta_{d_i}) = c(t,d_i) / \sum_{t' \in d_i} c(t',d_i)$. This equation assigns a high probability to a term if it has high probability in one of the document models, but not others, marginalizing over all the document models. This way, the higher the probability is, the more specific the term will be. Now, the goal is to extract the θ_c^{vs}. An EM algorithm, similar to Equations 5 and 6 can be applied for estimating θ_c^{vs} by removing the effect of θ_d^s from θ_c.

Considering the above process, a vertically separable model is a model which is **generalized** by taking out specific terms that have high probability in one of the descendant documents, but not others.

3.3 Two-Dimensional Separability

In order to have fully separable models in hierarchical classification, they should own two-dimensional separation property. We define two-dimensional separability as follows:

DEFINITION 4. *The model of an entity in the hierarchy is "two -dimensionally separable" if, and only if, it is both horizontally and vertically separable at the same time.*

Intuitively, if a model of an entity is two-dimensionally separable, it should capture *all*, and *only*, the essential features of the entity taking its relative position in the hierarchy in to consideration. In the next section, we will discuss how to estimate two-dimensional separable models for entities in the hierarchies with more than three layers.

In summary, based on the discussions on this section, we can say that the separation property is a desirable foundational property for classifiers. We see that based on PRP, separation in the feature space follows by separation in the score space, which leads to improvement in classification accuracy. We also notice that separation property in hierarchies is defined in two dimensions, thus, fully separable hierarchical models would possess both horizontal and vertical separation.

4. Hierarchical Significant Words Language Models

In this section, we address our second research question: "How can we estimate horizontally and vertically separable language models for the hierarchical entities?" We introduce Hierarchical Significant Words Language Models (HSWLM) which is an extension of Significant Words Language Model proposed by Dehghani et al. [11] to be applicable for hierarchical data. HSWLM is in fact a particular arrangement of multiple passes of the procedures of making hierarchical entities' models vertically and horizontally separable, as they are explained in Section 3.2. Generally speaking, hierarchical significant words language models are an extension of parsimonious language models [17] tailored to text-based hierarchical entities. In parsimonious language model, given a raw probabilistic estimation, the goal is to re-estimate the model so that non-essential parameters of the raw estimation are eliminated with regard to the background estimation. The proposed approach for estimating hierarchical significant words language model iteratively reestimates the standard language models of entities to minimize their overlap by discarding non-essential terms from them.

In the original parsimonious language model [17], background model is explained by the estimation of the *collection model*, i.e. the

Specification Stage

```
1: procedure SPECIFICATION
2:     Queue ← all entities in breadth first order
3:     while Queue is not empty do
4:         e ← Queue.pop()
5:         l ← e.Depth()
6:         while l > 0 do
7:             A ← e.GETANCESTOR(l)
8:             PARSIMONIZE(e,A)
9:             l ← l − 1
10:        end while
11:    end while
12: end procedure
```

(a) Procedure of Specification. e.GETANCESTOR(l) gives the ancestor of entity e with l edges distance from it.

Generalization Stage

```
1: procedure GENERALIZATION
2:     Stack ← all entities in breadth first order
3:     while Stack is not empty do
4:         e ← Stack.pop()
5:         l ← e.Height()
6:         while l > 0 do
7:             D ← e.GETDECEDENTS(l)
8:             PARSIMONIZE(e,D)
9:             l ← l − 1
10:        end while
11:    end while
12: end procedure
```

(b) Procedure of Generalization. e.GETDECEDENTS(l) gives all the decedents of entity e with l edges distance from it.

Figure 5: Pseudo-code for stages of estimating HSWLM. Function PARSIMONIZE(e,B) parsimonizes θ_e toward background models in B

model representing all the entities, similar to Equation 3. However, with respect to the hierarchical structure, and our goal in HSWLM for making the entities' models separable from each other, we need to use parsimonization technique in different situations: 1) toward ancestors of an entity, and 2) toward its descendants. Hence, beside parsimonizing toward a single parent entity in the upper layers, as the background model, we need to be able to do parsimonization toward multiple descendants in the lower layers. Figure 3 presents pseudo-code of Expectation-Maximization algorithm which is employed in the modified model parsimonization procedure. In the equation in line 3 of the pseudo-code in Figure 3, B is the set of background entities—either one or multiple, and θ_{b_i} demonstrates the model of each background entity, b_i, which is estimated using MLE. As can be seen, in case of having a single ancestor node as the background model, this equation will be equal to Equation 3 and in case of having multiple descendants as the background models, it results same as Equation 8. In this procedure, in general, in the E-step, the probabilities of terms are adjusted repeatedly and in the M-step, adjusted probability of terms are normalized to form a distribution. Another change in the modified version of model parsimonization, which practically makes no difference in the final estimation, is that in the E-step, instead of using $tf(t,e)$, we employ $p(t|\theta_e)$, where θ_e is the model represents entity e and initially it is estimated using MLE. This is because in the multi-layer hierarchies, there are more than one parsimonization pass for a particular entity and after the first round, we need to use the probability of terms estimated from the previous pass, not the raw information of their frequency.

Model parsimonization is an almost parameter free process. The only parameter is the standard smoothing parameter λ, which controls the level of parsimonization, so that the lower values of λ result in more parsimonious models. The iteration is repeated a fixed number of times or until the estimates do not change significantly anymore.

The pseudo-code of overall procedure of estimating HSWLM is presented in Figure 4. Before the first round of the procedure, a standard estimation like maximum likelihood estimation is used to construct the initial model for each entity in the hierarchy. Then, models will be updated in an iterative process until all the estimated models of entities become stable. In each iteration, there are two main stages: a *Specification stage* and a *Generalization stage*. In these stages, language models of entities in the hierarchy are iteratively made "specific," by taking out terms explained at higher levels, and "general," by eliminating specific terms of lower layers, which results in models that are both *horizontally* and *vertically* separable as it is described in Section 3.2.

In the specification stage, the goal is to eliminate the general terms of the language model of each entity so that the resulted language model demonstrates entity's specific properties. To do so, the parsimonization method is used to parsimonize the language model of an entity towards its ancestors, from the root of the hierarchy to its direct parent, as the background estimations. The order in the hierarchy is of crucial importance here. When a language model of an ancestor is considered as the background language model, it should demonstrate the "specific" properties of that ancestor. Due to this fact, it is important that before considering the language model of an entity as background estimation, it has already passed the specification stage, and we have to move top-down. Pseudo-code of the recursive procedure of specification of entities' models in the hierarchy is depicted in Figure 5a.

In the generalization stage, the goal is to refine language models by removing terms which do not address the concepts in the level of abstraction of the entity's layer. To do so, again parsimonization is exploited but towards descendants, which leads to elimination of specific terms. Here also, before considering the model of an entity as the background estimation, it should be already passed the generalization stage, so generalization moves bottom up. Figure 5b presents the pseudo-code for the recursive procedure of generalization of entities' language models in the hierarchy. In the generalization step, the background models of descendants are supposed to be specific enough to show their extremely specific properties. Hence, generalization stages must be applied on the output models of specification stages: specification should precede generalization, as shown in Figure 4 before.

In this section, we explained the procedure of estimating hierarchical significant words language model, which terminates in the language models that capture *all*, and *only* the essential terms regarding the hierarchical positions of entities. We will investigate the effectiveness of HSWLM on the real data in the next section.

5. EXPERIMENTS

In order to evaluate the separability of Hierarchical Significant Words Language Models, we use parliamentary data as one of the interesting collections with hierarchically structured data, using the hierarchical entities as shown in Figure 1. First we introduce the collection we have used and then we analyse the quality of HSWLM on providing horizontal and vertical separability over the hierarchy.

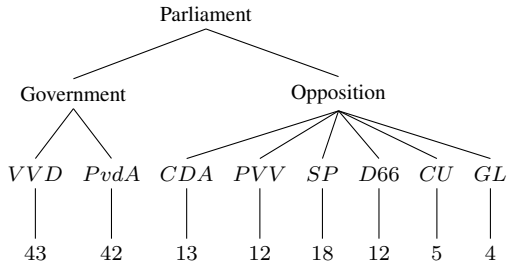


Figure 6: Composition of house of representatives of Dutch parliament, 2012-2014. *VVD*:People's Party for Freedom and democracy, *PvdA*:Labour Party, *CDA*:Christian Democratic Appeal, *PVV*:Party for Freedom, *SP*:The Socialist Party, *D66*:Democrats 66, *GL*:Green-Left, *CU*:Christian-Union

5.1 Experimental Dataset

We have made use of the Dutch parliamentary data to do a number of experiments. As a brief background, the Dutch parliament is a bicameral parliament which consists of the senate and the house of representatives. The house of representatives is the main chamber of parliament, where discussion of proposed legislation and review of the government's actions takes place. The Dutch parliamentary system is a multi-party system, requiring a coalition of parties to form the government [9]. For the experiments and analysis, we have used the data from the House of Representatives of Netherlands, consisting of literal transcripts of all speeches in parliament with rich annotation of the speaker and debate structure. We have chosen the last periods of parliament where eight main parties have about 95 percent of the 150 seats in the parliament. This data collected from March 2012 to April 2014 and consist of 62,206 debates containing 3.7M words. Figure 6 shows the hierarchical structure of house of representatives in this period. For each member, all speeches are collected from the discussions in the parliament and members for which the length of all their given speeches is less than 100 words are removed from the data instances. No stemming and no lemmatization is done on the data and also stop words and common words are not removed in data preprocessing.

5.2 Two-Dimensional Separability of HSWLM

In this section we investigate the ability of HSWLM on providing language models for hierarchical entities that are two-dimensionally separable. Based on the explained procedure of estimating HSWLM, the language models of entities in the hierarchy is repeatedly updated, so that the resulting models are both *horizontally* and *vertically* separable in the hierarchy. In order to assess this fact, we estimate HSWLM on the parliamentary data and look into the separability between entities in the same layer or in different layers.

Figures 7a and 7b illustrate the probability distribution over terms based on the estimated HSWLM in the status and party layer respectively. We sort the probability distribution on the term weight of the first model, and plot the other models in this exact order. As can be seen in the status layer, Figures 7a, the distributions over terms for government and opposition cover almost separated set of terms. Since in this layer these two entities are supposed to be against each other, a high level of separability can be expected. On the other hand, in the party layer, Figures 7b, it is possible that two parties share some ideological issues and consequently share some terms. So, in this layer a complete separability of terms would not be practically possible for all the parties. Nevertheless, HSWLM provides an acceptable horizontal separability in this layer.

In addition, we illustrate the horizontal separability of HSWLM of some pairs of parties. Figures 8a, 8b, and 8c show the separability of models of two parties in three cases, respectively: 1) different statuses, 2) both in the status of opposition, 3) both in the status of government. It can be seen that in all cases of being in the same status or different status, estimated hierarchical significant words language models are separable. The interesting point is in Figure 8c that presents the models of two government parties that are strongly separable. This rooted in the fact that in this period there was an unusual coalition government consisting of a right-wing and a left -wing party. So, although they have agreement in the status layer, their model is highly separable in terms of having opposite spectrum in party layer.

In order to illustrate the vertical separability of HSWLM, we choose two different branches in the hierarchy: one from leader of one of the opposition parties to the root, and the other from leader of one of the government parties to the root. Figures 9a and 9b show probability distributions over words based on HSWLM of all entities in these two branches. They demonstrate that using HSWLM, we can decompose distribution over all terms to the highly separable distributions, each one representing the language usage related to the meaning behind the layer of the entity in the hierarchy.

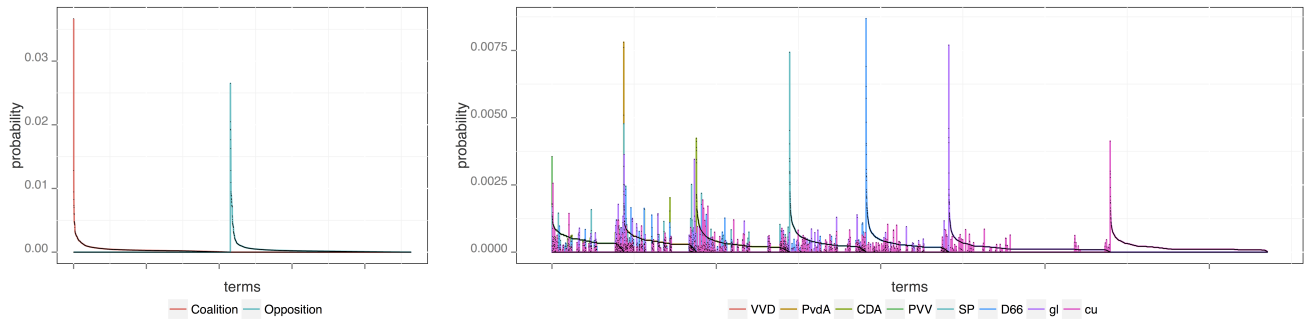
Two-dimensional separation property of HSWLM in the hierarchy is essentially due to the parsimonization effect in two directions. Intuitively, the horizontal separability is mainly the result of specification stage. For example, when an entity is parsimonized toward its direct parent, since the data in its parent is formed by pooling the data from the entity and its siblings, parsimonization makes the model of the entity separable from its siblings, which provide *horizontal separation* in the resulting language models. On the other hand, vertical separability is mainly due to generalization stage (and implicitly specification). For example, when an entity is parsimonized towards its children, since they are specified already, parsimonization gets rid of the specific terms of the lower layer from the entity's model.

5.3 Separability for Transferability

Here, we address our third research question: "How separability improves transferability?"

To address this question, we investigate the effectiveness of the models in the cross period classification task in the parliamentary dataset, which is to predict the party that a member of the parliament belongs to, having all the speeches given by that member in a period, as well as all the speeches given by the members of all parties in a different period of parliament. In the parliament, the status of the parties may change over different periods. Since the speeches given by the members are considerably affected by the status of their party, a dramatic change may happen in the parties' language usage. Due to this fact, learning a transferable model for party classification over periods is a very challenging task [18, 38].

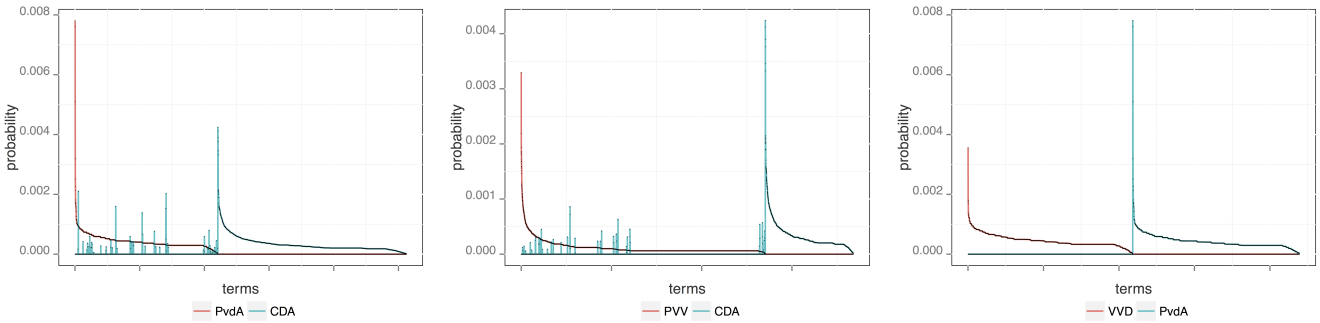
To evaluate the transferability of the models, besides the debates from the last period of Dutch parliament, we have used debates from October 2010 to March 2012 where VVD and CDA were pro-government parties and others were oppositions. We use SVM as the base classifier to predict party that each member belongs to, give the speaches of the members. We have done classification using the SVM itself as well as using SVM by considering probabilities of terms in HSWLM as the weights of features in order to evaluate the effectiveness of HSWLM as the separable representation of data. This way, we make use of HSWLM like a feature selection approach that filters out features that are not essential in accordance to the hierarchical position of entities and make the data representation more robust by taking out non-stable terms. We have also tried SVM along with other feature selection methods [5, 14] as the baselines, here we report the results of using Information Gain (IG) as the best feature selection method in our task in the parliament



(a) HSWLM in the status layer

(b) HSWLM in the party layer

Figure 7: *Horizontal Separability*: probability distribution over terms based on hierarchical significant words language models in status layer and party layer.

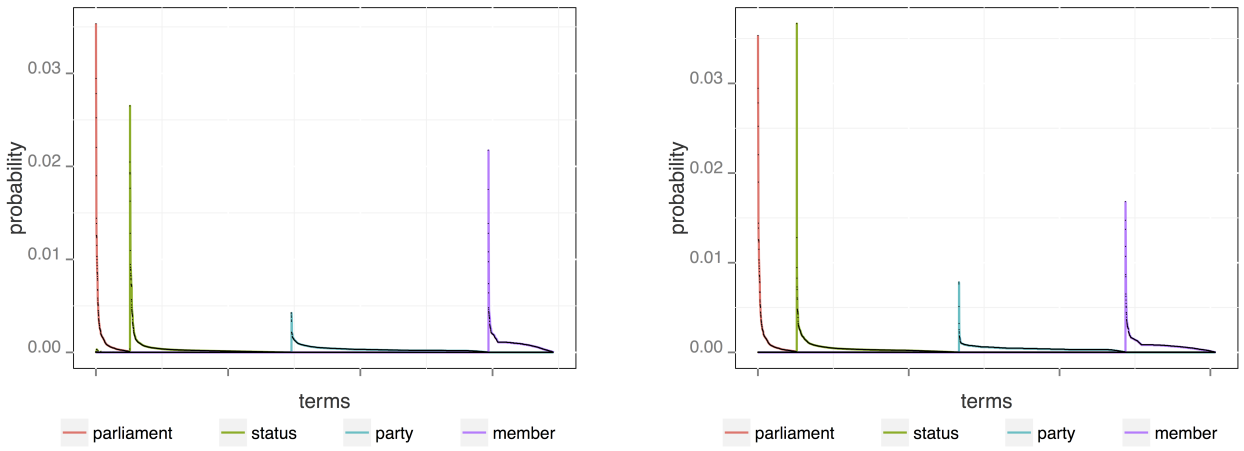


(a) HSWLM of two parties in different statuses: Christian Democratic Appeal (CDA) and Labour Party (PvdA)

(b) HSWLM of two parties in opposition: Party for Freedom (PVV) and Christian Democratic Appeal (CDA)

(c) HSWLM of two parties in government: People's Party for Freedom (VVD) and Labour Party (PvdA)

Figure 8: *Horizontal Separability*: probability distribution over terms based on hierarchical significant words language models in party layer



(a) HSWLM of S. van Haersma Buma (as the member of parliament - Leader of CDA), Christian Democratic Appeal (as the party), Opposition (as the status), and the Parliament

(b) HSWLM of D. Samson (as the member of parliament - Leader of PvdA), Labour Party (as the party), Government (as the status), and the Parliament

Figure 9: *Vertical Separability*: probability distribution over terms in different layers based on hierarchical significant words language models in complete paths from the root to the terminal entities in the hierarchy

Table 1: Results of party classification task in terms of macro-average accuracy. We have conducted paired t-test to investigate statistical significance of the improvements of the best method over the second best method, in the corresponding experiments. Improvements that are annotated with ▲ are statistically significant with p-value < 0.005.

		Test					
		SVM		SVM_{IG}		SVM_{HSWLM}	
	Period	2010-2012	2012-2014	2010-2012	2012-2014	2010-2012	2012-2014
Train	2010-2012	40.90	35.57	**43.11** ▲	34.12	41.83	**40.02**▲
	2012-2014	30.51	44.96	30.38	47.18	**39.11**▲	**47.28**

dataset. We have employed conventional 5-fold cross validation for training and testing and to maintain comparability, we have used the same split for folding in all the experiments. Tables 1 shows the performance of SVM, SVM_{IG}, and SVM_{HSWLM} on party classification over two periods in terms of macro-average accuracy. Comparing the results, it can be seen that SVM_{HSWLM} improves the performance of classification over SVM in all the experiments.

Although SVM_{IG} performs very well in terms of accuracy in within period experiments, it fails to learn a transferable model in cross period experiments and even it performs a little bit worse than the SVM itself. We looked into the confusion matrices of cross-period experiments and observed that most of the errors in both SVM and SVM_{IG} are because of misclassified members of CDA to PvdA and vice versa. These are the two parties that their statuses have been changed in these periods.

We investigate models of these two parties to understand how separation in the feature representation affects the performance of cross period classification. To do so, for each of these two classes, in each period, we extract three probability distributions on terms indicating their importance based on different weighting approaches: 1) Term Frequency (used as feature weights in SVM), 2) Information Gain (used as feature weights in SVM_{IG}), and 3) probability of terms in HSWLM (used as feature weights in SVM_{HSWLM}). Then, as a metric to measure separability of features, we use the Jensen-Shannon divergence to calculate diversity of probability distributions in three cases: 1) Different Parties in the Same Period, 2) Same Party in Different Periods 3) Different Parties in Different Periods. To avoid the effect of the number of features on the value of divergence, we take the top 500 high scored terms of each of weighting methods as the fixed length representatives of them. Figure 10 shows the average diversity of distributions in each of the three cases for each of the three weighting methods.

As expected, the diversity of features for different parties in a same period is high for all the methods and IG provides the more separable representation in this case, which results in its high accuracies in within period experiments. However, when we calculate the diversity of features for a same party in different periods, feature representations are different in both TF and IG, which causes false negative errors in the classification of these two parties. An interesting observation is in the case of having different parties in different periods, while we have two different parties their feature representations are similar in both TF and IG, which leads to false positive errors in the classification.

Considering these observations together reveals that SVM and SVM_{IG} learn models on the basis of features that are indicators of issues related to the status of parties, since they are the most discriminating terms considering one period and in within period experiments, the performance of SVM_{IG} and SVM is indebted to the separability of parties based on their statuses. Hence, after changing the status in the cross period experiments the trained model of the previous period generated by SVM and SVM_{IG} fails

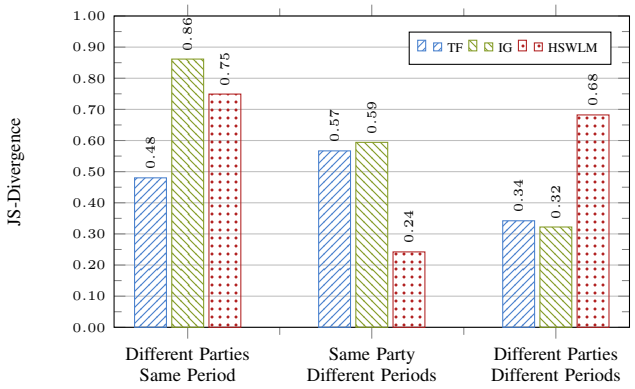


Figure 10: Average diversity of the representation of features of CDA and PvdA in different situations

to predict the accurate party. In the same way, the status classifier is affected by different parties forming a government in different periods, leading to lower accuracies.

This is exactly the point which the strengths of HSWLM kicks in. In fact, two-dimensional separability in the feature representation, enables SVM to tackle the problem of having non-stable features in the model when the status of a party changes over time. In other words, eliminating the effect of the status layer in the party model, which is the result of the horizontal separation, ensures that the party model captures the terms related to the party ideology, not its status. Thereby, not only SVM_{HSWLM} learns an effective model with acceptable accuracy in within period experiments, but also its learned models remain valid when the statuses of parties change.

We furthermore looked into the size of estimated HSWLMs by the number of terms with non-zero probability and, on average, the size of the models are about 100 times smaller than the number of features selected by IG in the corresponding models. So, although HSWLM takes considerable risk of loosing accuracy in within period experiments by aggressively pruning the overlapping terms, it provides small and precise models that are not only effective over time, but also efficient when the size of data is large.

In summary, we demonstrated that HSWLM indeed exhibit the two-dimensional separation property for the parliamentary data, as predicted by our theoretical analysis in the earlier sections. In addition, we empirically validated the transferability of models using HSWLM which is the result of its two-dimensional separability.

6. CONCLUSIONS

In this paper, we investigated the separation property in hierarchical data focusing on hierarchical text classification.

Our first research question was: "*What makes separability a desirable property for classifiers?*" We demonstrated that based on the ranking and classification principles, the *separation property* in the data representation is a desirable foundational property which leads to separability of scores and consequently improves the accuracy

of classifiers' decisions. We stated this as the "Strong Separation Principle" for optimizing expected effectiveness of classifiers.

Our second research question was: "*How can we estimate horizontally and vertically separable language models for the hierarchical entities?*" We showed that in order to have horizontally and vertically separable models, they should capture all, and only, the essential terms of the entities taking their position in the hierarchy into account. Based on this, we introduced Hierarchical Significant Words Language Models for estimating separable models for hierarchical entities. We investigated HSWLM and demonstrated that it offers separable distributions over terms for different entities both in case of being in the same layer or in different layers.

Our third research question was: "*How separability improves transferability?*" We evaluated the performance of classification over time using separable representation of data and showed that separability makes the model more robust and transferable over time by filtering out non-essential non-stable terms.

The models we proposed in this paper are IR models which are applicable to a range of information access tasks [10–13], not just hierarchical classification, as many complex ranking models combine different layers of information. There are number of extensions we are working on in future work. First, in this research we focused on text-dominant environment and considered all terms in the text as features. However, this can be done considering terms with a specific part of speech, or even on non-textual models with completely different types of features. Second, it would be beneficial to construct mixture models for terminal entities using HSWLM in a way that constructed mixture models are capable of reflecting local interactions of terminal entities in different layers.

Acknowledgments This research is funded in part by Netherlands Organization for Scientific Research through the *Exploratory Political Search* project (ExPoSe, NWO CI # 314.99.108), and by the Digging into Data Challenge through the *Digging Into Linked Parliamentary Data* project (DiLiPaD, NWO Digging into Data # 600.006.014).

References

[1] Feature generation and selection for information retrieval. Workshop of SIGIR, 2010.

[2] A. Arampatzis and A. van Hameran. The score-distributional threshold optimization for adaptive binary classification tasks. In *SIGIR '01*, pages 285–293, 2001.

[3] A. Arampatzis, J. Kamps, and S. Robertson. Where to stop reading a ranked list?: Threshold optimization using truncated score distributions. In *SIGIR '09*, pages 524–531, 2009.

[4] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[5] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Feature selection using linear support vector machines. Technical Report MSR-TR-2002-63, Microsoft Research, 2002.

[6] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[7] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *NIPS '24*, pages 2456–2464. 2011.

[8] F. Crestani, M. Lalmas, C. J. Van Rijsbergen, and I. Campbell. "is this document relevant?... probably...": A survey of probabilistic models in information retrieval. *ACM Comput. Surv.*, 30(4):528–552, Dec. 1998.

[9] A. de Swaan. *Coalition Theories and Cabinet Formations: A Study of Formal Theories of Coalition Formation Applied to Nine European Parliaments after 1918*, volume 4 of *Progress in Mathematical Social Sciences*. Elsevier, New York, 1973.

[10] M. Dehghani. Significant words representations of entities. In *SIGIR '16*, pages 1183–1183, 2016.

[11] M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *CIKM '16*, 2016.

[12] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Generalized group profiling for content customization. In *CHIIR '16*, pages 245–248, 2016.

[13] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Two-way parsimonious classification models for evolving hierarchies. In *CLEF '16*, 2016.

[14] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

[15] S. Gopal and Y. Yang. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *SIGKDD*, pages 257–265, 2013.

[16] V. Ha-Thuc and J.-M. Renders. Large-scale hierarchical text classification without labelled data. In *WSDM*, pages 685–694, 2011.

[17] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *SIGIR*, pages 178–185, 2004.

[18] G. Hirst, Y. Riabinin, J. Graham, and M. Boizot-Roche. Text to ideology or text to party status? *From Text to Political Positions: Text analysis across disciplines*, 55:93–15, 2014.

[19] E. Kanoulas, V. Pavlu, K. Dai, and J. Aslam. Modeling the score distributions of relevant and non-relevant documents. In *ICTIR'09*, volume 5766, pages 152–163. Springer Berlin Heidelberg, 2009.

[20] D.-k. Kim, G. Voelker, and L. K. Saul. A variational approximation for topic modeling of hierarchical corpora. In *ICML*, pages 55–63, 2013.

[21] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, 2001.

[22] D. D. Lewis. *Representation and Learning in Information Retrieval*. PhD thesis, Amherst, MA, USA, 1992.

[23] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR '95*, pages 246–254, 1995.

[24] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*, pages 359–367, 1998.

[25] P. Ogilvie and J. Callan. Hierarchical language models for xml component retrieval. In *INEX*, pages 224–237, 2004.

[26] H.-S. Oh, Y. Choi, and S.-H. Myaeng. Text classification for a large-scale taxonomy using dynamically mixed local and global models for a node. In *ECIR*, pages 7–18, 2011.

[27] S. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977.

[28] T. Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. *JASIST*, 26:321–343, 1975.

[29] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[30] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An element-based approach to xml retrieval. In *INEX*, pages 19–26, 2004.

[31] Y. Song and D. Roth. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585, 2014.

[32] K. Sparck Jones, H. D. Robertson, Stephen, and Z. Hugo. Language modeling and relevance. In *Language Modeling for Information Retrieval*, pages 57–71. 2003.

[33] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. In *ICDM*, pages 521–528, 2001.

[34] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[35] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *SIGIR '08*, pages 627–634, 2008.

[36] G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *SIGIR*, pages 619–626, 2008.

[37] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *SIGKDD*, pages 937–946, 2009.

[38] B. Yu, S. Kaufmann, and D. Diermeier. Classifying party affiliation from political speech. *Journal of Information Technology & Politics*, 5 (1):33–48, 2008.

[39] E. Zavitsanos, G. Paliouras, and G. A. Vouros. Non-parametric estimation of topic hierarchies from texts with hierarchical dirichlet processes. *J. Mach. Learn. Res.*, 12:2749–2775, 2011.

[40] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, 2001.

[41] D. Zhou, L. Xiao, and M. Wu. Hierarchical classification via orthogonal transfer. In *ICML*, pages 801–808, 2011.