Two-Way Parsimonious Classification Models for Evolving Hierarchies

Mostafa Dehghani[1], Hosein Azarbonyad[2], Jaap Kamps[1], and Maarten Marx[2]

[1] Institute for Logic, Language and Computation, University of Amsterdam
[2] Informatics Institute, University of Amsterdam
{dehghani,h.azarbonyad,maartenmarx,kamps}@uva.nl

Abstract. There is an increasing volume of semantically annotated data available, in particular due to the emerging use of knowledge bases to annotate or classify dynamic data on the web. This is challenging as these knowledge bases have a dynamic hierarchical or graph structure demanding robustness against changes in the data structure over time. In general, this requires us to develop appropriate models for the hierarchical classes that capture all, and only, the essential solid features of the classes which remain valid even as the structure changes. We propose *hierarchical significant words language models* of textual objects in the intermediate levels of hierarchies as robust models for hierarchical classification by taking the hierarchical relations into consideration. We conduct extensive experiments on richly annotated parliamentary proceedings linking every speech to the respective speaker, their political party, and their role in the parliament. Our main findings are the following. First, we define hierarchical significant words language models as an iterative estimation process across the hierarchy, resulting in tiny models capturing only well grounded text features at each level. Second, we apply the resulting models to party membership and party position classification across time periods, where the structure of the parliament changes, and see the models dramatically better transfer across time periods, relative to the baselines.

Keywords: Hierarchical Significant Words Language Models, Evolving Hierarchies.

1 Introduction

Modern web data is highly structured in terms of containing many facts and entities in a graph or hierarchies, making it possible to express concepts at different levels of abstraction. However, due to the dynamic nature of data, their structure may evolve over time. For example, in a hierarchy, nodes can be removed or added or even transfer across the hierarchy. Thus, modeling objects in the evolving structures and building robust classifiers for them is notoriously hard and requires employing a set of solid features from the data, which are not affected by these kinds of changes.

For example, assume we would build a classifier for the "US president" over recent data, then a standard classifier would not distinguish the role in office from the person who is the current president, leading to obvious issues after the elections in 2016. In other words, if we can separate the model of the function from the model of the person fulfilling it, for example by abstracting over several presidents, that more general model would in principle be robust over time.

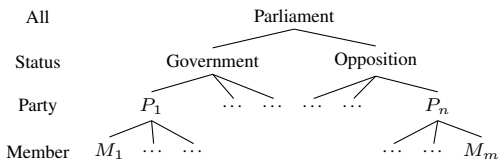

Fig. 1: Hierarchical relations in parliament.

These challenges are ubiquitous in dealing with any dynamic data annotated with concepts from a hierarchical structure. We study the problem in the context of parliamentary data, as a particular web data. Parliamentary proceedings in public government are one of the fully annotated data with an enriched dynamic structure linking every speech to the respective speaker, their role in the parliament and their political party.

Consider a simple hierarchy of a multi-party parliament as shown in Figure 1, which determines different categories relevant to different layers of membership in the parliament. Also assume that all speeches of members of the parliament are available and each object in the hierarchy is represented using all the speeches given by members affiliated by the object. It is desirable to use text classification approaches to study how speeches of politicians relate to ideology or other factors such as party membership or party status as government or opposition, over different periods of parliament. To this end, we need models representing each object in the intermediate levels of the hierarchy as a category representing all its descendant objects. However, in the parliament hierarchy, since members and parties can move in the hierarchy over different periods, it is challenging to estimate models that transfer across time. For instance, after elections, governments change and prior opposition parties may form the new government, and prior government parties form the new opposition. Thus, if the model of, say, status in terms of government and opposition, is affected by terms related to the parties' ideology, they will not be valid in the next period. This requires making these models less dependent on the "accidental" parties and members forming the government in a particular period and capture the essential features of the abstract notion of status.

In order to estimate a robust model for an object in an evolving hierarchy, we need to explicitly take all the relations between the object and other objects in other layers into account and try to capture essential features by removing features that are better explained by other objects in different layers. This way, by estimating independent models for related objects, we can assure that the models remain valid even if the relational structure of the hierarchy changes over time.

Based on this, we propose *hierarchical significant words language models* (HSWLM) of hierarchical objects, which are highly robust against structural changes by capturing, all, and only the significant terms as stable set of features. Our inspiration comes from the early work on information retrieval by Luhn [13], in which it is argued that in order to establish a model consisting of *significant words*, we need to eliminate both common words and rare words. Based on this idea, with respect to the structure of the hierarchy, we propose to define general terms as terms already explained by ancestor models, and specific terms as terms already explained by models of descendants, and then employ the

parsimonization technique [10] to hierarchically eliminate them as non-essential terms from the models, leading to models that capture permanent significant words.

The main aim of this paper is *to develop appropriate language models for classification of objects in the evolving hierarchies.* We break this down into a number of concrete research questions:

1. How to estimate robust language models for objects in the evolving hierarchies, by explicitly taking relations between the levels into account?
2. How effective are hierarchical significant words language models for classifying textual objects regarding different levels of the hierarchy across time periods?
3. Do the resulting hierarchical significant words language models capture common characteristics of classes in different levels of hierarchy over time?

The rest of the paper is structured as follows. Next, in Section 2, we discuss related work. Section 3 introduces our approach to estimate hierarchical significant words language models. In Section 4 we apply our models to the parliamentary proceedings, and show how effective are HSWLMs to model party status and party membership across different government periods. Furthermore, we investigate the ability of models for capturing similar and stable features of parliamentary objects over time. Finally, Section 5 concludes the paper and discusses extensions and future work.

2 Related Work

There is considerable research related to our work in terms of using the same type of data, or focusing on the problem of hierarchical text classification or aiming on improving transferability of models over time, which we discuss them in this section.

There is a range of work on political data which is related to our research in terms of using the same type of data and hierarchical structure. The recent study of Hirst et al. [11] is the closest to our work. They presented an analytical study on the effectiveness of classifiers on political texts. Using Canadian parliamentary data they demonstrated that although classifiers may perform well in terms of accuracy on party classification in the parliamentary data, they pick the expressions of opposition and government, of attack and defence, or of questions and answers, and not of ideology. They also showed that using classic approach for categorization fails in extracting ideology by examining the models over different government periods. In our paper, we examine our method also with the evaluation strategy of Hirst et al., and in contrast to the failure of classic categorization methods on parliamentary data reported before, we demonstrate that our proposed method performs well under these difficult conditions.

Although our research problem differs from issues in typical hierarchical text classification problems using a topical hierarchy [8, 9, 19, 20], we review some research in this area and will use effective approaches like SVM as baselines in our experiments. McCallum et al. [15] proposed a method for modeling an object in the hierarchy, which tackles the problem of data sparseness for low layered objects. They used shrinkage estimator to smooth the model of each leaf object with the model of its ancestors to make them more reliable. Ogilvie and Callan [16] and Oh et al. [17] extended the McCallum et al.'s idea by including the models of children as well as parents, and controlling the level of information that is needed to be gathered from ancestors. Recently, Song and Roth [21] tackled the problem of representing hierarchical objects with the lack

of training data by embedding all objects in a semantic space to be able to compute a meaningful semantic similarity between them. Although the general problem in these papers is similar to ours, they address the problem of *train data sparseness* [15, 21] or present techniques for *handling large scale data* [17].

In terms of modeling hierarchical objects, there are similarities with work on hierarchical topic modeling. Kim et al. [12] used Hierarchical Dirichlet Process [HDP, 22] to construct models for objects in the hierarchies using their own models as well as the models of their ancestors. Also Zavitsanos et al. [26] used HDP to construct the model of objects in a hierarchy employing the models of its descendants. These research try to bring out precise topic models using the structure of the hierarchy, but they do not aim to capture a model which keeps its validity over the time even while changes occur in the structural relations. The longitudinal changes in the data in our problem, relate it to the works on constructing dynamic models for data streams [1, 24]. In this line of research, they 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 remain valid over time. Research on domain adaptation [2, 23] also tried to tackle the problem of missing features when very different vocabulary are used in test and train data. This differs from our approach first in terms of considering the hierarchical relations, and also the fact that we aim to estimate models that are robust against changes in the structural relations, not the corpus vocabulary.

3 Significant Words Language Models

In this section, we address our first research questions: "How to estimate robust language models for objects in the evolving hierarchies, by explicitly taking relations between the levels into account?" We propose to extract hierarchical significant words language models (HSWLM) as models estimated for objects in evolving hierarchies that are robust and *persistent* even by changing the structural relations in the hierarchy over time. Each object in the hierarchy is assumed to be a textual document, representing the corresponding concept of that object in the hierarchy.

Basically, our proposed approach, two-way parsimonization, tries to iteratively re-estimate the models by discarding non-essential terms from them. This pruning for each object is accomplished using parsimonization technique toward both the ancestors of the object and its descendants. One of the main components of the process of estimating HSWLM is the procedure of *Model Parsimonization*, which we will discuss first.

3.1 Model Parsimonization

Model parsimonization is a technique that was introduced by Hiemstra et al. [10] in which given a raw probabilistic estimation, the goal is to re-estimate the model so that non-essential terms are eliminated with regard to the background estimation.

To do so, each term t in the object model, θ_o, assumed to be drawn from a two-component mixture model, where the first component is the background language model, θ_B, and the other is the latent parsimonious model of the object, $\tilde{\theta}_o$. With regard to the generative models, when a term t is generated using this mixture model, first a model is chosen and then the term is sampled using the chosen model. Thus, the probability of generating term t can be shown as follows:

$$p(t|\theta_o) = \alpha p(t|\tilde{\theta}_o) + (1-\alpha)p(t|\theta_B), \tag{1}$$

Model Parsimonization

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

(a) Pseudo-code for EM procedure of parsimonization.

Estimating HSWLM

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

(b) Pseudo-code for procedure of estimating HSWLM.

Specification Stage

procedure SPECIFICATION
  Queue ← all objects in BFS order
  while Queue is not empty **do**
    $o \leftarrow$ Queue.pop()
    $l \leftarrow o$.Depth();
    while $l > 0$ **do**
      $A \leftarrow o$.GETANCESTOR(l)
      PARSIMONIZE(o,A)
      $l \leftarrow l - 1$
    end while
  end while
end procedure

(c) Procedure of Specification. o.GETANCESTOR(l) gives the ancestor of object o with l edges distance from it.

Generalization Stage

procedure GENERALIZATION
  Stack ← all objects in BFS order
  while Stack is not empty **do**
    $o \leftarrow$ Stack.pop()
    $l \leftarrow o$.Height();
    while $l > 0$ **do**
      $D \leftarrow o$.GETDECEDENTS(l)
      PARSIMONIZE(o,D)
      $l \leftarrow l - 1$
    end while
  end while
end procedure

(d) Procedure of Generalization. o.GETDECEDENTS(l) gives all the decedents of object o with l edges distance from it.

Fig. 2: Pseudo-code of Estimating hierarchical significant words language models.

where α is the standard smoothing parameter that determines the probability of choosing the parsimonious model to generate the term t. The log-likelihood function for generating all terms in the whole object o is:

$$\log p(o|\tilde{\theta}_o) = \sum_{t \in o} c(t,o) \log \left(\alpha p(t|\tilde{\theta}_o) + (1-\alpha) p(t|\theta_B) \right), \tag{2}$$

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

E-step:

$$p[t \in \mathcal{T}] = c(t|o).\frac{\alpha p(t|\tilde{\theta}_o)}{\alpha p(t|\tilde{\theta}_o) + (1-\alpha) p(t|\theta_B)}, \tag{3}$$

M-step:

$$p(t|\tilde{\theta}_o) = \frac{p[t \in \mathcal{T}]}{\sum_{t' \in \mathcal{T}} p[t' \in \mathcal{T}]} \tag{4}$$

where $\mathcal{T}$ is the set of all terms with non-zero probability in the initial estimation. In Equation 3, θ_o is the maximum likelihood estimation. $\tilde{\theta}_o$ represents the parsimonious model, which in the first iteration, is initialized by the maximum likelihood estimation, similar to θ_o.

Modified Model Parsimonization In the original model parsimonization [10], the background model is explained by the estimation of the *collection language model*, i.e. the model representing all the objects. So, according to Equation 3, parsimonization penalizes raw inference of terms that are better explained by the collection language model, as the background model, and continuing the iterations, their probability is adjusted to zero. This eventually results in a model with only the specific and distinctive terms of the object that makes it distinguishable from other objects in the collection.

However, with respect to the hierarchical structure, and our goal in two-way parsimonization for removing the effect of other layers in the object model, we need to use parsimonization technique in different situations: 1) toward ancestors of the object 2) toward its descendants. Hence, besides parsimonizing toward a single parent object in the upper layers, as the background model, we need to be able to do parsimonization toward multiple descendants in lower layers.

We propose the following equation for estimating the background model, which supports multiple background object, to be employed in the two-way model parsimonization:

$$p(t|\theta_B) \xleftarrow{normalized} \sum_{t_i \in B} \left(p(t|\theta_{b_i}) \prod_{\substack{b_j \in B \\ j \neq i}} (1 - p(t|\theta_{b_j})) \right) \tag{5}$$

In this equation, B is the set of background objects—either one or multiple, and θ_{b_i} demonstrates the model of each background object, b_i, which is estimated using MLE. We normalize all the probabilities of the terms to form a distribution.

In two-way parsimonization, regarding the abstraction level in the hierarchy, when the background model represents an ancestor object in the upper layers of the hierarchy, it is supposed to reflect the generality of terms, so that parsimonaizing toward this model brings "specification" for the estimated model by removing general terms. On the other hand, when the background model represents multiple descendants from lower layers, it is supposed to reflect the specificity of terms, so that parsimonaizing toward this model brings "generalization" for the estimated model by discarding specific terms.

According to the aforementioned meanings of background model in these situations, Equation 5 provides a proper estimation: In the multiple background case, it assigns a high probability to a term if it has a high probability in one of the background (descendant) models but not others, marginalizing over all the background models. This way, the higher the probability is, the more specific the term will be. In the single background case, i.e having only one background object in the set B, $p(x|\theta_B)$ would be equal to $p(x|\theta_b)$, i.e MLE of background object b. Since this single background object is from upper layers that are more general, this model reflects generality of terms.

Figure 2a presents pseudo-code of Expectation-Maximization algorithm which is employed in the modified model parsimonization 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.

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

3.2 Estimating HSWLM

We now investigate the question: How hierarchical significant words language models provide robust models by taking out aspects explained at other levels? In order to estimate HSWLM, in each iteration, there are two main stages: a *Specification stage* and a *Generalization stage*. In loose terms, in the specification stage, the model of each object is specified relative to its ancestors and in generalization stage, the model of each object is generalized considering all its descendants. The pseudo-code of overall procedure of estimating HSWLM is presented in Figure 2b. Before the first round of the procedure, a standard estimation like maximum likelihood estimation is used to construct the initial model for each object in the hierarchy. Then, in each iteration, models are updated in specification and generalization stages. These two stages are repeated until all the estimated models of all objects become stable.

In the specification stage, the parsimonization method is used to parsimonize the model of an object toward its ancestors, from the root of the hierarchy to its direct parent, as background estimations. The top-down order in the hierarchy is important here. Because when a model of an ancestor is considered as the background estimation, it should demonstrate the "specific" properties of that ancestor. Due to this fact, it is important that before considering the model of an object as the background estimation in specification stage, it should be already specified toward its ancestors. Pseudo-code for the recursive procedure of specification of objects' model is shown in Figure 2c.

After specification stage, unless the root object, the models of all the objects are updated and the terms related to general properties are discarded from all models. In the generalization stage, again parsimonization is exploited but toward descendants. In the hierarchy, descendants of an object are usually supposed to represent more specific concepts compared to the object. Although the original parsimonization essentially accomplishes the effect of specification, parsimonizing the model of an object toward its descendants' models means generalizing the model. Here also, before considering the model of an object as background estimation, it should be already generalized toward its ancestors, so generalization moves bottom up. Figure 2d presents the pseudo-code for the recursive procedure of generalization of objects' model. It is noteworthy that the order of the stages is important. In the generalization, 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 as shown in Figure 2b where specification precedes generalization.

It is noteworthy that although the process of estimating HSWLM is an iterative method, it is highly efficient. This is because of the fact that in the first iteration of the process, model parsimonization in specification and generalization stages results in tiny effective models which do not contain unessential terms. Therefore, in the next iterations, the process deals with sparse distributions, with very small numbers of essential terms.

In this section, we proposed to iteratively use of parsimonization to take out general aspects explained at higher levels and estimate more specific and precise models as well as eliminating specific aspects of lower layers, to make models more general, —resulting in hierarchical significant words language models.

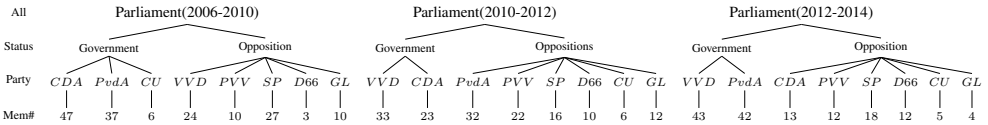

Fig. 3: Composition of Dutch parliament in 3 periods. *VVD*:People's Party for Freedom and democracy, *PvdA*:Labour Party, *CDA*:Christian Democratic Appeal, *PVV*:Party for Freedom, *SP*:The Socialist Party, *D66*:Democrats 66, *GL*:Green-Left, *CU*:Christian-Union.

4 HSWLM for Evolving Hierarchies

This section investigates our second research question: "How effective are hierarchical significant words language models for classifying textual objects regarding different levels of the hierarchy across time periods?" We first explain the data collection we used as well as our experimental settings. Then we discuss how the estimation method addresses the requirement outlined in the introduction.

4.1 Data Collection and Experimental Settings

In this research, we have made use of the Dutch parliamentary data. The data are collected and annotated as the part of *PoliticalMashup* project [18] to make semantically enriched parliamentary proceedings available as open data [14].

As a brief background, Dutch parliamentary system is a multi-party system, requiring a coalition of parties to form the government. We have chosen three interesting periods of parliament, from March 2006 to April 2014, in which eight main parties have about 95% of seats in the parliament. The coalition in the first period is between a left-wing party and a centrism party, in the second period between a right-wing party and centrism party, and in the third, between a right-wing and left-wing party. Figure 3 shows the hierarchical structure of Dutch parliament in these three different periods.

In order to model parliamentary objects, first of all, we prepare the data. In the proceedings, there are series of parliamentary speeches by different MPs following the debate structure. We invert the data matrix so that for each speaker we collect their speeches as a single document, which represents the features of that member. Then, for representing the internal objects in the parliament's hierarchy, we first consider members as the leaf objects and then concatenate all leaf documents below internal objects as a single document which textually represent them: first over parties, and then parties into government and opposition, etc. The whole corpus consists of 14.7 million terms from 240,501 speeches, and contains 2.1 million unique terms. No stemming and no lemmatization is done on the data and also stop words and common words are not removed in data preprocessing. After data preparation, we estimate HSWLM for all objects in the hierarchy as it is explained in Section 3.

4.2 Classification across Periods

As an extrinsic evaluation of the estimated models, we investigate the question: "How hierarchical significant words language models provide robust models by taking out aspects explained at other levels?" In the parliament, the composition of parties and statuses changes over different periods (Figure 3) and hence the speeches related to different objects can vary dramatically. Due to this fact, cross period classification is notoriously challenging [11, 25]. We show that our proposed approach tackles the

Table 1: Results on the task of status classification.

(a) Accuracy of SVM classifier

Period		Test			
		2006-10	2010-12	2012-14	All
Train	2006-10	84.14	68.83	87.24	-
	2010-12	68.29	78.57	87.91	-
	2012-14	68.90	75.97	88.59	-
	All	-	-	-	79.87

(b) Accuracy of classifier uses HSWLM

Period		Test			
		2006-10	2010-12	2012-14	All
Train	2006-10	82.32	80.51	89.29	-
	2010-12	79.87	74.66	88.58	-
	2012-14	78.65	77.27	93.28	-
	All	-	-	-	86.98

Table 2: Results on the task of party classification.

(a) Accuracy of SVM classifier

Period		Test			
		2006-10	2010-12	2012-14	All
Train	2006-10	47.56	29.22	26.84	-
	2010-12	29.87	40.90	35.57	-
	2012-14	31.09	30.51	44.96	-
	All	-	-	-	39.18

(b) Accuracy of classifier uses HSWLM

Period		Test			
		2006-10	2010-12	2012-14	All
Train	2006-10	44.51	46.10	43.62	-
	2010-12	40.85	40.25	39.59	-
	2012-14	40.24	38.96	42.28	-
	All	-	-	-	49.94

problem of having non-stable models when the composition of parliament evolves during the time, by capturing the essence of language models of parliamentary objects at aggregate levels.

Tables 1b and 2b show the performance of employing HSWLM on status and party classification respectively. As a hard baseline, we have employed SVM classifier on parliamentary data like experiments done in [7] and also examined it on the cross period situation. Tables 1a and 2a indicate the results of SVM classifier on status and party classification respectively. Comparing the results in Tables 1b and 1a, we see that the accuracy of SVM in within period experiments is sometimes slightly better, but in cross period experiments, classifier which uses HSWLM of statuses achieves better results. This is also observed in the results in Table 2b compare to the results in Table 2a.

For party classification, employing HSWLM results more significant improvement over the baseline. Hirst et al. [11] discuss that since the status of members in parliament, compare to their party, has more effect on the content of their speeches, classifiers tend to pick features related to the status, not the party ideologies. So, SVM performs very well in terms of accuracy in the within-period experiments, but this performance is indebted to the separability of parties due to their status. Hence, changing the status in cross period experiments, using trained model on other periods fails to predict the party so the accuracies drop down. This is exactly the point which the strengths of our proposed method kicks in. Since for each party, the HSWLM is less affected by the status of the party in that period, the model remains valid even when the status is changed. In other words, eliminating the effect of the status layer in the party model in the specification stage ensures that party model captures the essential terms related to the party ideology, not its status. Thereby, it is a stable model which is transferable through the time. We conducted the one-tailed t-test on the results. In both party and status classification, in all cases which HSWLM performs better than the SVM, the improvement is statistically significant (p-value < 0.005).

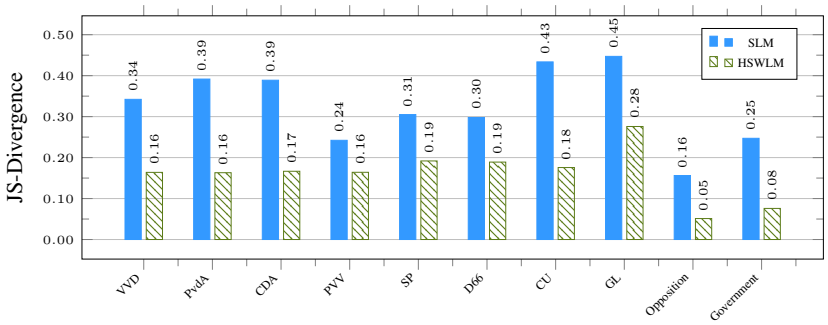

Fig. 4: Average of JS-Divergence of standard language models and HSWLMs for parliamentary entities in three different periods.

To get a better intuition of the procedure of estimating HSWLM, consider the hierarchical relations of Dutch parliaments in the period of *2006-2010* which is depicted in Figure 3. Assume that the goal is modeling language usage of "Christian-Union (CU)" as an object in the party layer. In the speeches from the members of this party, words like "*Chairman*" or "*Agree*" might occur repeatedly. However, they are not a good point of reference for the party's ideological language usage. In the procedure of estimating HSWLM of the "Christian-Union", these words are removed from the initial estimated standard language model in the specification stages, since "*Chairman*" is a general term in the parliamentary domain and is only able to explain the root object and "*Agree*' is somehow an indicator of language usage of all the "Government" parties. On the other side, consider the goal is to model language usage of "Government" as an object in the status layer. Speeches from "Christian-Union" members, which are also counted as "Government" members, may contain words like "*Bible*" or "*Charity*". It is trivial that involving these party-specific words in the constructed model for the "Government" in an individual period demolishes the comprehensiveness. In the procedure of estimating HSWLM for the "Government", in the generalization stages, these words are discarded from the model. This way, "Government" model does not lose its validity on other periods where the "Christian-Union" is not in a Government party.

As another indicator of the effectiveness of HSWLM, it outperforms the SVM bringing all the data together from three different periods in both party and status classification. This is because it gets the chance of having a more rich train data which leads to more precise models. While in SVM, changes in the parliamentary composition make speeches diverse and this makes it not to be able to learn a concrete model.

4.3 Invariance of Models

This section investigates our third research question: "Do the resulting hierarchical significant words language models capture common characteristics of classes in different levels of hierarchy over time?" As an intrinsic evaluation of the models, we evaluate the invariance of models over different periods—how similar are models of a particular object in the hierarchy when trained on data from different periods. Since HSWLM is supposed to captures the essence of objects, not only HSWLM of an object learned using an individual period should be valid for representing the object on other periods, but also models of the same object learned on data from different periods should be invariant.

To assess this, we use the diversity of objects' models in different periods to measure their (in)variance over time. First, all HSWLM from different periods of each party and each status is smoothed using Jelinek-Mercer smoothing [27] considering all parliamentary speeches in the corresponding period as the background collection and with the same value of the smoothing parameter. Then, we use the Jensen-Shannon divergence as the diversity metric to measure dissimilarities between each two HSWLMs learned from different periods and then calculate the average of values for each object. As the baseline, the same calculation is done for the standard language models of objects, i.e language models estimated using maximum likelihood estimation. Figure 4 shows the diversity of models in different periods. As can be seen, in all objects in both party and status layers, diversity of HSWLM of different periods is lower than diversity of standard language models, which shows the extracted HSWLMs are more invariant over different periods.

In this section, we examined classification accuracy over time using HSWLM and saw significantly better results across different government periods. This suggest that HSWLM captures the essential and permanent features of parliamentary objects. Moreover, we looked at the divergence of models from different periods, and observed that HSWLMs from different periods are more invariant compared to the standard models.

5 Conclusions

In this research, we dealt with the problem of modeling hierarchical objects for building classifiers in different levels of evolving hierarchies. To address this problem, inspired by parsimonious language models used in information retrieval, we proposed *hierarchical significant words language models* (HSWLM).

Our first research question was: *How to estimate robust language models for objects in the evolving hierarchies, by explicitly taking relations between the levels into account?* We proposed the iteratively use of parsimonization to take out general aspects explained at higher levels and eliminate specific aspects of lower levels—resulting in HSWLM. Our second question was: *How effective are hierarchical significant words language models for classifying textual objects regarding different levels of the hierarchy across time periods?* We utilized HSWLM for the task of party and status classification in the parliament over time. The results showed that since the models capture the essential and permanent features of parliamentary objects, they lead to significantly better classification accuracy across different government periods. Our third question was: *Do the resulting hierarchical significant words language models capture common characteristics of classes in different levels of hierarchy over time?* We designed an experiment in which divergence of models from different periods is measured for all objects. We observed that HSWLMs from different periods are more consistent compared to the standard models.

The general idea of HSWLM is to estimate models possessing separation property [6] and it is applicable in other problems [3–5]. Besides, we are currently extending the work in this paper in several directions.

First, we apply the approach to other kinds of web data in particular social network data. Second, we investigate the effectiveness of the models for various other hierarchical classification tasks, in particular those over dynamic or stream data, and develop variants dealing with data sparsity. Third, we further develop new variants of topic models building on the specialization and generalization outlined in this paper.

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).

6 References

[1] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.
[2] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *NIPS '24*, pages 2456–2464. 2011.
[3] M. Dehghani. Significant words representations of entities. In *SIGIR '16*, 2016.
[4] M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *The proceedings of The ACM International Conference on Information and Knowledge Management (CIKM'16)*, 2016.
[5] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Generalized group profiling for content customization. In *CHIIR '16*, pages 245–248, 2016.
[6] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. On horizontal and vertical separation in hierarchical text classification. In *The proceedings of ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR'16)*, 2016.
[7] D. Diermeier, J.-F. Godbout, B. Yu, and S. Kaufmann. Language and ideology in congress. *British Journal of Political Science*, 42(1):31–55, 2012.
[8] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR*, pages 256–263, 2000.
[9] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. M. Voorhees, and I. Soboroff. Evaluating stream filtering for entity profile updates in trec 2012, 2013, and 2014. In *TREC*, 2014.
[10] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *SIGIR*, SIGIR '04, pages 178–185, 2004.
[11] 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.
[12] D.-k. Kim, G. Voelker, and L. K. Saul. A variational approximation for topic modeling of hierarchical corpora. In *ICML*, pages 55–63, 2013.
[13] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, 1958.
[14] M. Marx and A. Schuth. Dutchparl: A corpus of parliamentary documents in dutch. In *DIR Workshop*, pages 82–83, 2010.
[15] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*, ICML '98, pages 359–367, 1998.
[16] P. Ogilvie and J. Callan. Hierarchical language models for xml component retrieval. In *INEX*, pages 224–237, 2005.
[17] 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.
[18] PoliticalMashup. Political mashup project. `http://search.politicalmashup.nl/`, 2015. Netherlands Organization for Scientific Research.
[19] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34 (1):1–47, Mar. 2002.
[20] C. N. Silla, Jr. and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.*, 22(1-2):31–72, 2011.
[21] Y. Song and D. Roth. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585, 2014.

[22] 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.

[23] 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.

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

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

[26] 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.

[27] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, SIGIR '01, pages 334–342, 2001.