Explaining Classes through Stable Word Attributions

Samuel Rönnqvist, Amanda Myntti, Aki-Juhani Kyröläinen
Filip Ginter and **Veronika Laippala**
TurkuNLP
University of Turku, Finland
first.last@utu.fi

Abstract

Input saliency methods have recently become a popular tool for explaining predictions of deep learning models in NLP. Nevertheless, there has been little work investigating methods for aggregating prediction-level explanations to the class level, nor has a framework for evaluating such class explanations been established. We explore explanations based on XLM-R and the Integrated Gradients input attribution method, and propose 1) the Stable Attribution Class Explanation method (SACX) to extract keyword lists of classes in text classification tasks, and 2) a framework for the systematic evaluation of the keyword lists. We find that explanations of individual predictions are prone to noise, but that stable explanations can be effectively identified through repeated training and explanation. We evaluate on web register data and show that the class explanations are linguistically meaningful and distinguishing of the classes.

1 Introduction

In recent years, various approaches to explaining predictions of deep neural networks have been attracting interest in the fields of NLP and computer vision (see, Montavon et al. (2018)). Several techniques have been suggested in this vein, including model attention visualization (see, e.g., Vig and Belinkov (2019)), and input attribution (or saliency) methods (see Bastings and Filippova, 2020; Ding and Koehn, 2021; Simonyan et al., 2014), which focus on explaining individual predictions. However, showing how a model perceives larger units such as entire classes in a text classification task would be crucial for gaining a global understanding of deep classifiers and salient word features.

Moreover, text classification models often struggle to truly generalize (Laippala et al., 2021; Petrenz and Webber, 2011). For instance, McCoy et al. (2020) show in repeated experiments with BERT on a text inference task that, while consistent test set performance was achieved, the degree of generalization as measured on a related task varied significantly, due to randomized initializations of the decision layer and order of training examples. Similarly, Laippala et al. (2021) demonstrate that resampling of the data had a positive impact on feature stability of linear support vector machines. Thus, various random aspects of the training process may affect the reliability of modeling results, beyond predictive performance on a test set, especially in deep language models.

In this paper, we propose a method for explaining classes in a text classification task using deep language models based on input attributions estimated with the Integrated Gradients (IG) method (Sundararajan et al., 2017). We focus specifically on IG as it provides a general framework for estimating feature importance in deep neural networks and has been shown to provide reliable saliency maps in text classification among other tasks. For a discussion on the merits of IG, cf. Prasad et al. (2021), and Bastings and Filippova (2020) on saliency vs. attention methods in general.

Our class explanation method works by aggregating attributions in two ways: across documents and across models. On the one hand, we classify documents and aggregate word attribution scores from them, in order to extract the overall most predictive word features of a particular class. On the other hand, we aggregate these attributions over multiple random train/validation data splits and instances of a classifier, in order to identify stable attributions that are consistently assigned across rounds. Thus, we consider the level of a particular classifier configuration—i.e., the combination of language model, decision layer, hyperparameters, loss function, etc.—and strive to capture its perception of a corpus.

Our method explains a class in the form of a list of words ranked by the aggregated attribution

Findings of the Association for Computational Linguistics: ACL 2022, pages 1063 - 1074

scores, and filtered based on their stability across experiments. Following corpus linguistics' long tradition of analyzing style and content of text classes, we refer to these attributions as keywords (see Scott and Tribble, 2006; Stubbs, 2010, for discussion). This type of analysis is concerned with identifying the words that are most informative about the characteristics conveyed by a given text class.

While keyword analysis is widely employed in corpus linguistics, quantitative measures have been used only for extraction and not as a framework for evaluation, which is rather done qualitatively (cf. Egbert and Biber, 2019). Thus, as a contribution of this paper, we propose three lexical measures of keyword quality, which help us optimize and evaluate our method. We also study syntactic and semantic properties to nuance our understanding of keywords obtained with a deep classifier and IG.

We test our method by training a set of classifiers on the Corpus of Online Registers of English (CORE) (Egbert et al., 2015). CORE is sampled from the searchable English-language web and aims to be representative of the distribution of registers (or genres) found online. Recent work in both linguistics and NLP has, however, demonstrated challenges of categorizing language use on the web pertaining to its extreme variation within and across classes (Titak and Robertson, 2013; Dayter and Messerli, 2021; Madjarov et al., 2019; Biber and Egbert, 2019). Therefore, explanation methods are especially needed in web register classification, in order to explore the robustness and linguistic motivation of blackbox models.

We put forward our *Stable Attribution Class Explanation* method (SACX)[1] as a support in understanding classes and their modeling by deep language model classifiers, in text classification tasks where keywords provide a suitable means of explanation. It can assist model development and debugging by highlighting salient word features, at a more general level compared to attributions at the document and classifier instance level.

2 Data

CORE (Egbert et al., 2015) is a large-scale collection of web texts annotated for their genre, or register (Biber, 1988). In total, the dataset consists of nearly 50,000 texts. In our experiments, we combine the train and development sets, totaling 38,760 texts. The CORE register classes are coded using a two-level taxonomy developed in a data-driven manner to cover the full range of web language use. We focus on the upper level which consists of eight register classes: Narrative (NA), Opinion (OP), How-to (HI), Interactive discussion (ID), Informational description (IN), Lyrical (LY), Spoken (SP) and Informational persuasion (IP). Additionally, the dataset includes hybrid documents featuring characteristics of several registers and thus coded with several register labels (see Table 5 in Appendix).

3 Methods

3.1 Classifier and attribution method

As a classifier, we use the XLM-R deep language model (Conneau et al., 2020) because of its strong ability to model multiple languages, both in monolingual and cross-lingual settings. We opt for the base size rather than the large, due to its relatively frugal use of resource and comparable predictive performance on CORE (Repo et al., 2021). The task is modeled as a multilabel classification task using a sequence classification head, binary cross-entry with sigmoid loss and a fixed prediction threshold. We optimize the classifier hyperparameters against the development set, in order to reuse the settings in the explanation process described below.

We use the IG method to obtain explanations from the XLM-R predictions[2] (Sundararajan et al., 2017). IG takes the network input in the form of token embeddings and a corresponding blank reference input (same-length sequence of embeddings for a fixed placeholder token), and calculates a linear interpolation between them over a number of steps (e.g., 50). It then calculates gradients to measure the relationship between changes in an embedding and changes in the model predictions. This produces attribution scores for each dimension of the input token embeddings. Our explanation method then aggregates these in several steps into class representations.

3.2 The Stable Attribution Class Explanation method (SACX)

The class descriptions are extracted through the steps detailed below and illustrated in Figure 1.

[1]The code is available at: https://github.com/TurkuNLP/class-explainer/

[2]We use the Huggingface transformers library for modeling and the Captum implementation of IG (Kokhlikyan et al., 2020).

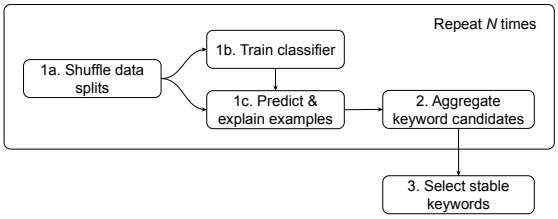

Figure 1: Overview of the SACX method.

Step 1: Train and explain. We combine the training and development sets of the corpus and randomly split them into a new training and validation set according to a set ratio r, using stratification to keep class distributions stable (cf. Laippala et al., 2021). The pre-trained language model is loaded and the decision layer is randomly initialized. Both are fine-tuned on the new training set. Documents in the validation set are classified by a threshold τ on the posterior probabilities, and the IG method is applied in order to obtain attribution scores for the network inputs, i.e., each dimension of each input token embedding, w.r.t. each predicted class c.

Step 2: Aggregate attributions from documents. The attribution scores for each embedding dimension are summed up per token to provide a token-level score, while all tokens in a document d are normalized by the L_2 norm. This provides a word attribution score $s_{w,d,c}$ directly if the word w consists of a single token, otherwise it is calculated as the maximum of all sub-word token scores. We calculate the average attribution scores $\bar{s}_{w,c}$, for each (w, c), as a means for ranking the keywords for each class. In order to reduce noise, we only select the n top-scoring words per document d, and we only consider true positive predictions. We note that the method could alternatively be used for error analysis by targeting false predictions.

Step 3: Select stable keywords. The above process is repeated N times, each time randomly shuffling and splitting the data and reinitializing the classification head according to Step 1, in order to quantify the stability of the keywords. The keyword candidates ranked by $\bar{s}_{w,c}$ are filtered based on selection frequency: a word is considered stable if the ratio by which it is selected (in Step 2) across the experiments is larger than a threshold value t.

Finally, we perform a light cleaning by ignoring words that occur in less than k documents and do not contain any alphabetic characters. We optimize the parameters t, n and τ in the experiments.

3.3 Baseline methods

We use the two following methods for extraction of class keywords, as baselines in comparison:

TF-IDF. As a naïve approach, we create a TF-IDF model with logarithmic scaling, a minimum document frequency of 10 and a maximum document frequency at 50% of the number of documents in the largest class. To extract the keywords, a class vector is formed by first averaging the document vectors for a given class from the weight matrix and then taking the 100 highest scoring terms as keywords for each class.

SVMs. As a strong baseline, we follow Sharoff et al. (2010); Laippala et al. (2021). We use a linear Support Vector Machine (SVM) with L2 penalty and TF-IDF vectorizer with a minimum document frequency of 0.05%, in Scikit-learn (LinearSVC). SVMs were adapted to the multilabel setting using a one-versus-rest strategy, and the C value optimized with grid search (0.5 providing the best scores). We train the SVMs on the same random splits as XML-R. During each round, the 1000 best positive features for each class are extracted. For the selection of the stable keywords, a selection frequency threshold of 0.6 was chosen.

4 Evaluation setting

We evaluate the keyword quality based on usefulness and relevance, which are established concepts in feature selection and evaluation in machine learning (e.g., Blum and Langley, 1997; Kohavi and John, 1997; Guyon and Elisseeff, 2003). Usefulness refers to the discriminative power of the features used in a task, e.g., as measured by how well they allow to discriminate the classes in a test set. Relevance refers to the association of the features with the actual object of study, i.e., their generalizability beyond a test set. Not all useful features are relevant—for instance, some useful features may inherit their usefulness from data idiosyncrasies, unrepresentative train/test splits and spurious statistics (see Ribeiro et al., 2016). In the case of keywords, useful keywords allow to discriminate the classes in the data, while relevant keywords reflect meaningful and linguistically motivated characteristics associated with the classes.

We propose three measures for assessing usefulness of keywords based on lexical overlap, presented in Section 4.1, which we use to optimize parameters of our explanation method and to compare against the baseline methods. In Section 4.2,

we present further analyses conducted to assess the relevance of keywords and to form a qualitative understanding of the differences in output of the methods. The results are presented in Section 5.

4.1 Lexical measures of usefulness

Our measures related to usefulness focus on 1) distinctiveness—how distinct or overlapping keywords are between classes, 2) coverage—how well the keywords cover the documents of the corpus, and 3) a combination of the two that measures distinctiveness based on coverage. Similar to previous studies, we only consider the top-100 keywords (see Pojanapunya and Todd, 2018).

4.1.1 Distinctiveness (intrinsic)

We first propose a simple intrinsic measure, which assesses the distinctiveness of keywords, by looking at keyword overlap. Specifically, it measures the fraction of keywords unique to a class, averaged across classes:

$$Dist_{int} = \frac{1}{|C|} \sum_{c \in C} \frac{|\{k | k \in K_c \setminus K_{\neg c}\}|}{|K_c|}$$

for the set of classes C and keywords K for class c or all other classes $\neg c$. Whereas keyword analysis tends to focus on binary categories and methods that separate keyword by design, our measure fits more general uses, e.g., in settings with multiple classes.

4.1.2 Coverage

In the next step, we look at lexical coverage of the keywords in associated documents in the corpus as an indicator of usefulness. We define coverage of a class as the average proportion of keywords that occur across all its documents, and the global coverage measure as the macro average across all classes:

$$Cov = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|T_c|} \sum_{t \in T_c} \frac{|\{k | k \in K_c \cap t\}|}{|K_c|}$$

where T_c is the set of texts of class c, either based on true class membership or true positive predictions. We again focus on true positives as we are interested in evaluating the quality of the keywords relative to the learned representation, not factoring in the model's predictive performance.

4.1.3 Distinctiveness (extrinsic)

Having defined a measure of coverage, we derive an extrinsic measure of distinctiveness as the coverage of keywords within a class relative to the coverage across the class boundary, of unrelated documents. We define a cross-coverage measure:

$$XCov = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|T_{\neg c}|} \sum_{t \in T_{\neg c}} \frac{|\{k | k \in K_c \cap t\}|}{|K_c|}$$

of keywords K and texts $T_{\neg c}$, which is the set of texts not labeled with class c.

The extrinsic distinctiveness is then defined as:

$$Dist_{ext} = \frac{Cov - XCov}{Cov}$$

which provides an easy-to-interpret metric in the range $[0, 1]$, where a distinctiveness score of 0 means that there is no difference in keyword coverage within and across classes, and a score of 1 indicates a perfect separation between classes.

Egbert and Biber (2019) propose a similar notion of "content-distinctiveness" based on text dispersion keyness (incorporating document frequency) as a desirable quality of keywords.

4.2 Syntactic and semantic analysis of relevance

Our analysis of relevance focuses on syntactic and semantic properties associated with the keywords. While traditionally the relevance of keywords is assessed qualitatively and based on intuition (e.g., Scott and Tribble, 2006; Bondi and Scott, 2010; Gabrielatos and Marchi, 2011; Phillips, 1989; Williams, 1976), the goal of the proposed analysis is to provide inference for contrasting the three methods. This also allows us to deepen our understanding of the keywords.

First, we assess the proportion of content and function words among the keywords. This is an important qualitative distinction in keyword analysis, and generally methods extracting keywords with a stronger affinity to topicality/content rather than grammatical/functional elements are considered to be superior (cf. Egbert and Biber, 2019).

We parse the corpus with Turku Neural Parser (Kanerva et al., 2018), identify the most frequent part-of-speech (POS) per keyword, and group their distribution into two lexical categories: function and content words. Function words consist of adpositions, conjunctions, pronouns, auxiliaries, adverbs, interjections and determiners, and content words of adjectives, nouns, proper nouns and verbs. Other POS classes (numbers, symbols, punctuation, particles) are excluded from the analysis.

Class	F1 (*M*)	*SD*	Sup. (*M*)
Lyrical (LY)	82.28	8.78	180.58
Narrative (NA)	77.83	1.79	5779.71
Inter. discussion (ID)	75.67	3.06	915.35
Inform. description (IN)	65.73	1.42	3352.49
Opinion (OP)	55.41	5.08	2803.13
How-to (HI)	54.23	5.85	538.51
Inform. persuasion (IP)	43.83	6.14	527.80
Spoken (SP)	25.93	23.10	195.49
Micro AVG	68.37	1.84	–

Table 1: Predictive performance of XML-R classifier as mean F1-score (%), with standard deviation and mean support across the resampling rounds ($N = 100$).

Second, we examine the keywords from the perspective of semantic coherence. We analyze keyword similarities relative to the semantic structure of the corpus as a whole using word embeddings and clustering. We turn the dataset vocabulary into word vectors, using FastText vectors pre-trained on Common Crawl, 600B tokens in 300 dimensions (Mikolov et al., 2018). This ensures that the semantic vectors are independent from the explanation methods, while being trained on data similar to CORE, namely unrestricted web text. The analysis is further described in Section 5.5.

5 Results

After completing $N = 100$ rounds of experiments, we first inspect the predictive performances of the trained classifiers and study the degree of stability of the attributions. Then, we report the results of the optimization of our method against the usefulness-focused lexical measures, and qualitatively inspect the extracted top keywords. Finally, we present the syntactic and semantic analyses focusing on keyword relevance.

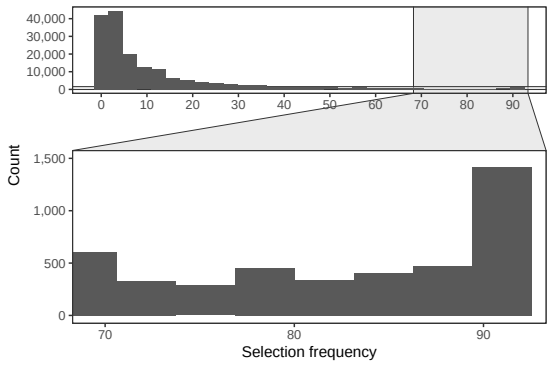

Figure 2: Distribution of selection frequency of keyword candidates for all classes, based on $N = 100$ rounds. The upper panel shows the full range and the lower the subrange of stable keywords.

5.1 Predictive performance

Table 1 summarizes the predictive performance of the 100 XLM-R classifiers that we have trained.[3] The micro average F1-score was a good 68% on average. Similar to previous studies (Repo et al., 2021; Rönnqvist et al., 2021; Biber and Egbert, 2016), we observed a large variation among classes, ranging from an F1-score of 44% (Informational persuasion) to 81% (Lyrical). Our method was able to extract stable keywords for all the classes expect for Spoken, where no keyword candidate passed the selection frequency threshold. This was mirrored both in its significantly lower class-specific F1-score of 26% and the exceptionally high standard deviation of 23%, likely related to the small sample size. For comparison, the SVMs baseline achieved a micro F1-score of 65.00% (*SD* = 0.33%).

5.2 Stability of keywords

We investigated the (in)stability of keywords across the 100 runs, and the utility of the selection frequency threshold t, by studying the selection frequency of the keyword candidates. The distribution of selection frequency is visualized in Figure 2. We see that the vast majority of keyword candidates appear only in a low number of runs.

For instance, for Informational Description exhibiting the lowest standard deviation in F1-score (1.42%), the top-10 unfiltered words were: *lollies, verdant, especially, endorsing, forebears, equations, gerald, colin, indy* and *exaggerating*. These keyword candidates scored in the range 0.79–0.93, but had selection frequencies of only 1 and 3 (for colin). By comparison, the first word with a selection frequency above $t = 0.7$ is *abstract* (selection frequency 98%), with a score of 0.56 which ranks it 45$^{\text{th}}$ before filtering (cf. Table 3). In fact, in order to extract the top-100 stable keywords we consider in evaluation, we need to traverse the unfiltered lists of keyword candidates, on average, down to rank 22,940 (range 1,775–67,859 for all classes). This illustrates the extent of instability among the attributions.

Finally, comparing the keywords extracted from the XLM-R and SVMs, we observed that the SVMs produced more consistent results with a mean selection frequency of 92.16% among the top-100 filtered keywords vs. 74.01% for the aggregations based on XLM-R and IG. This further highlights

[3]The optimal setting used was learning rate 7.5e-5 and batch size 30 for 12 epochs with early stopping (patience 1).

Method	Dist., intrinsic	Dist., extrinsic	Coverage
SACX	82.57	44.27	10.08
SVMs	91.43	39.94	10.53
TF-IDF	29.86	43.92	10.37

Table 2: Method comparison based on distinctiveness (dist.) and coverage (in %).

how inconsistent the attributions obtained by the IG method are across different runs, and thereby confirms the necessity of selection frequency filtering in obtaining stable and likely meaningful keywords.

5.3 Comparison on lexical usefulness measures

We apply the three lexical measures introduced in Section 4.1 to evaluate the keywords of our proposed method and the baselines. We set the parameters $r = 0.67$ (split), $N = 100$ (runs) and $k = 5$ (minimum document frequency), while optimizing the rest with grid search against the lexical measures. Weighting the three measures against each other is not entirely trivial, as they capture different qualities and we do not have a clear preference a priori. With different settings we are able to maximize different measures: intrinsic distinctivenss to 85.43%, extrinsic distinctiveness to 58.98%, and coverage to 11.33%.[4] However, maximizing either form of distinctiveness severely hurts coverage. We found a good balance with the settings $t = 0.7$ (selection frequency threshold), $n = 20$ (words per document) and $\tau = 0.7$ (prediction threshold), which achieves comparable coverage to the other methods and competitive numbers for the distinctiveness measures. The results for this setting and the baselines are listed in Table 2.

Based on intrinsic distinctiveness both our SACX method (83.6%) and SVMs (91.4%) displayed strong discriminative power, contrasting TF-IDF (29.9%). In terms of lexical coverage across the documents, all methods performed at the same level (10.08–10.53%). Similarly, the methods displayed a modest difference in performance based on extrinsic distinctiveness: SACX (44.3%) followed by TF-IDF (43.9%) and SVMs (39.9%). Taken together, the results demonstrate that the keywords extracted by our method were useful in discriminating between the classes, performing similarly to SVMs, while TF-IDF stood out with its weak separation of keywords across the classes.

5.4 Extracted keywords

The top-15 keywords of each class are presented in Table 3 and the keywords extracted with the baseline methods in Table 6 in Appendix.[5]

Our method was able to extract relevant keywords that clearly reflect our understanding of these seven classes and also share similarities with keywords discovered for these data in previous studies (e.g., Biber and Egbert, 2019; Laippala et al., 2021). The keywords are predominantly content words reflecting the class characteristics, such as *faq, question, answer, forum* extracted for Interactive discussion (ID). Similarly, linguistically-motivated patterns emerged from other classes, such as keywords associated with research papers and reports from Informational description (IN) (*abstract, introduction, summary, bio*) and keywords, in particular proper nouns, reflecting news and sports from Narrative (NA) (*afp, reuters* and *bundesliga, nba, ufc, playoffs, nfl, uefa, psg*).

The keywords extracted with the baseline methods are linguistically motivated as well. However, instead of extracting mainly content words, they identified also function words as keywords, such as *or, it, we, doesn* and *dont* (cf. Section 4.2). Many of these function words identified as keywords are linguistically motivated and reflect descriptions established in previous studies on register analysis (Biber, 1988; Biber and Egbert, 2016, 2019).

5.5 Analysis of relevance of the keywords

In our syntactic analysis, we evaluate relevance based on the relative frequencies of content and function words, as listed in Table 4. Relative to the baselines, SACX shows a tendency to extract less function and more content words, in particular more nouns (including proper nouns). This suggests that it is more likely to focus on topical keywords. The distributional differences were statistically significant ($X^2(8, N = 2,100) = 111.33, p < 0.001$) and a residual analysis confirmed the negative association with function words and the positive one with proper nouns.

In our semantic analysis, we visualize the full lexical space by reducing the 300 dimensions to two using Uniform Manifold Approximation and Projection (McInnes et al., 2018). The SACX key-

[4] The settings being in the same order: $t = (0.7, 0.4, 0.8)$, $n = (50, 40, 30)$, $\tau = (0.7, 0.5, 0.7)$. Tested ranges were $t = [0.3, 0.8]$, $n = [10, 50]$, $\tau = [0.5, 0.9]$.

[5] Full lists of keywords from all methods are available during review as supplementary material, online upon publication.

– How-to (HI) –			– Inter. Discussion (ID) –			– Inform. Description (IN) –			– Inform. Persuasion (IP) –		
Keyword	Score	SF	Keyword	Score	SF	Keyword	Score	SF	Keyword	Score	SF
how	0.5206	97	faq	0.5806	98	abstract	0.5633	98	description	0.5049	96
howto	0.4024	87	question	0.5514	98	storyline	0.4589	88	isbn	0.4181	70
diy	0.3538	77	answer	0.4815	98	faqs	0.4412	95	product	0.2804	96
recipe	0.3368	97	forum	0.4733	98	faq	0.4198	95	book	0.2776	96
recipes	0.2965	97	answers	0.4554	98	aspect	0.3403	97	important	0.2603	93
to	0.2425	96	thread	0.4202	98	introduction	0.3057	98	shop	0.2431	73
ingredients	0.2344	97	forums	0.4007	98	summary	0.3023	98	details	0.2322	93
tutorial	0.2311	96	re	0.3786	98	contents	0.3016	98	amazon	0.2131	96
tutorials	0.2268	78	discuss	0.3723	98	abstracts	0.2838	90	reviews	0.2034	96
tips	0.2194	97	answered	0.3645	93	bio	0.2635	92	buy	0.1807	96
tip	0.2012	94	replies	0.3554	98	disclaimer	0.2538	98	available	0.1787	96
navigation	0.1910	77	threads	0.3490	98	meta	0.2519	74	review	0.1772	96
remove	0.1870	97	resolved	0.3284	98	profiles	0.2500	86	item	0.1732	76
build	0.1849	91	quote	0.3280	98	downloads	0.2471	72	package	0.1711	70
preheat	0.1831	87	answerer	0.3188	98	dictionary	0.2441	98	products	0.1681	96

– Lyrical (LY) –			– Narrative (NA) –			– Opinion (OP) –		
Keyword	Score	SF	Keyword	Score	SF	Keyword	Score	SF
lyrics	0.4117	97	bundesliga	0.3588	98	review	0.5456	98
poem	0.2839	75	afp	0.3462	98	weblog	0.5028	72
written	0.1583	81	nba	0.3455	98	psalm	0.4444	95
sorry	0.1471	70	ufc	0.3327	98	feminist	0.3376	94
lyricsmode	0.1462	91	blog	0.3307	98	tips	0.3292	92
truth	0.1351	91	playoffs	0.3283	98	blog	0.3279	98
songs	0.1343	73	nfl	0.3263	98	bible	0.3250	98
yeah	0.1337	95	wordpress	0.3253	87	thursday	0.3000	98
tired	0.1331	79	flickr	0.3248	95	lgbt	0.2957	87
finally	0.1314	76	playoff	0.3075	98	eucharistic	0.2925	71
tonight	0.1312	87	reuters	0.3073	98	monday	0.2899	97
something	0.1302	97	uefa	0.3065	98	tuesday	0.2873	98
heaven	0.1300	77	zlatan	0.3055	98	wednesday	0.2861	98
lord	0.1299	74	psg	0.3038	97	testament	0.2780	98
fucking	0.1287	79	responses	0.3000	92	post	0.2688	98

Table 3: Top-15 extracted keywords for each class ranked by mean aggregated attribution score (Score). The lists are filtered by threshold on selection frequency (SF in %).

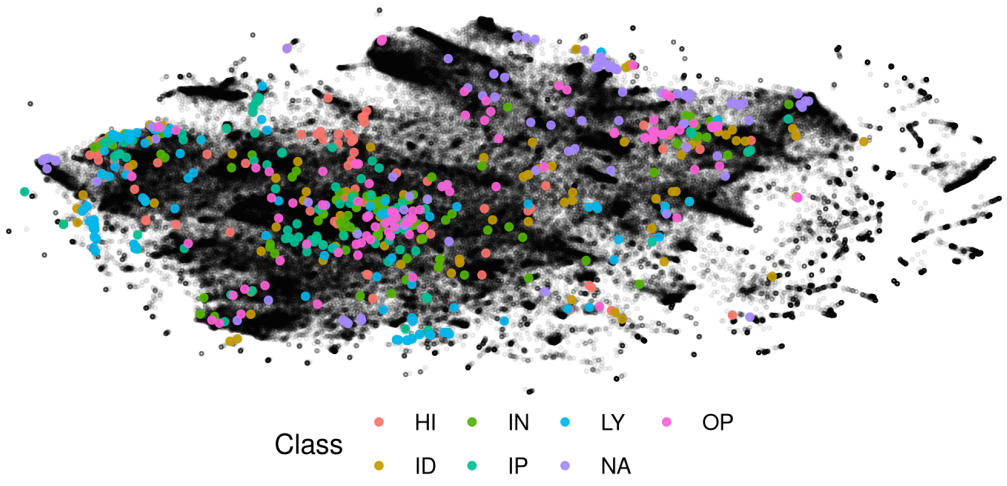

Figure 3: The lexical space of CORE, with keywords extracted from XLM-R colored based on class.

words are highlighted and colored by class, in Figure 3, and the baseline keywords in Figure 6 in Appendix. We observe that the SACX keywords cluster densely to a higher degree, suggesting semantically more coherent keywords.

To formally test this, we clustered the semantic vectors of the whole vocabulary using model-based clustering with mixtures from von Mises-Fisher distributions (Banerjee et al., 2005; Hornik and Grün, 2014) as the data were unit vectors. We

Method	Content word				Function
	Adj.	Noun	Prop.n.	Verb	
SACX	8.61	48.49	13.77	16.79	12.34
SVMs	12.10	49.86	6.34	15.99	15.71
TF-IDF	13.09	39.86	1.29	30.07	15.68

Table 4: Distribution of lexical classes of the keywords for each method (in %).

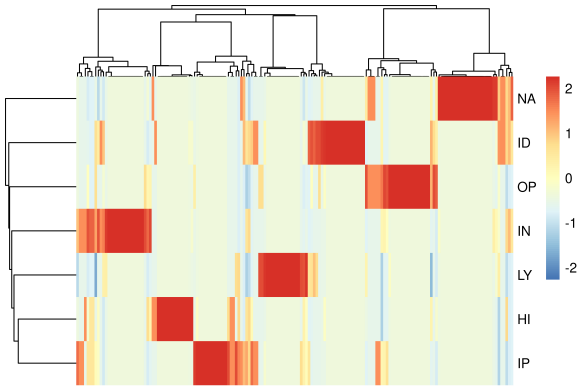

Figure 4: Cluster solution with clusters mapped to SACX keywords (columns) relative to the classes (rows). The color indicates the strength of association.

found 500 clusters to be optimal, based on BIC (Schwarz, 1978) and visual inspection indicating no substantive difference with fewer clusters.

Density was parametrized by the mean direction μ and the concentration parameter κ characterizing the strength of concentration of the data about the mean direction. This analysis showed that SACX was 1.25 times (OR 95% CIs = 1.03, 1.5) more likely to extract the keywords from dense clusters (above average k) than the other two methods together. Considering the previously noted propensity of SACX to extract proper nouns, we also studied their frequencies in dense vs. sparse clusters. We found that SACX was 5.85 times (OR 95% CIs = 3.07, 11.1) more likely to extract proper nouns from dense clusters than the other two methods together. This suggests that its keywords are both more specific and coherent in terms of vector space similarity.

Figure 4 visualizes the SACX keywords by the clusters they were assigned (columns) relative to the classes (rows), with a hierarchical biclustering on the axes. It further demonstrates the semantic coherence of the keywords as indicated clearly by the horizontal tightness and the strength of association (increase in redness). By comparison, in Figure 5 in Appendix, we see somewhat less coherence with SVMs, and clearly less with TF-IDF.

6 Conclusion

We have presented the Stable Attribution Class Explanation method (SACX) for explaining classes in text classification, based on IG input attributions from deep language model classifiers. SACX produces lists of keywords reflecting a classifier's perception of classes. However, input attributions are prone to noise, which we have shown can be effectively filtered, as we performed 100 rounds of training an XLM-R classifier and applying IG.

We have demonstrated that these *stable* keywords are of good quality—both useful as features and meaningfully relevant of the text classes studied. We have proposed lexical measures for evaluating distinctiveness and corpus coverage of keywords, and we have compared our method against two baseline class explanation methods. We compared the methods based on syntactic and semantic properties of the keywords, and found SACX to distinguish itself in that it extracts more content and less function words—a property which is generally considered to be a hallmark of a superior keyword analysis method in corpus linguistics. In particular, SACX has the ability to focus on more specific, topical words in the form of proper nouns, when relevant for depicting the class (such as for Narrative).

We have shown that SACX produces keywords that are highly coherent and tend to cluster densely throughout semantic vector space, rather than being evenly dispersed such as the word features extracted from SVMs. We also demonstrated that proper nouns are a distinguishing feature of these dense clusters, further illustrating the coherence of SACX keywords. We speculate that the use of token embeddings, and the XLM-R model's ability to learn local and highly non-linear functional forms afforded by the significant number of parameters, may give rise to these keyword characteristics.

In the future, we seek to explore the utility of the method in various settings, and further investigate the quality and nature of its class explanations. We will test it on further text classification tasks and types of models, as well as apply the approach to other languages and cross-lingual settings. In particular, understanding model behavior in zero-shot classification through stable explanations at the class level may provide a useful tool in detecting systematic biases. In the context of register identification, recent pursuits in this direction of multi- and cross-lingual modeling (Repo

et al., 2021; Rönnqvist et al., 2021; Laippala et al., 2019) have been making good progress in terms of predictive performance, but interpretability tools such as ours could offer linguistic insight, e.g., into language-independent markers of the classes.

Moreover, as we have demonstrated that input attributions are highly prone to noise at the level of individual classifier instances, the type of filtering we have proposed can be used, not only to stabilize class-level explanations, but, more generally to generate stable saliency maps for particular text inputs based on multiple classifier instances. Future work should explore this direction further, as the contextualized interpretation of individual text inputs can provide a useful complement to the keyword-based class explanations for understanding model behavior.

Acknowledgements

We thank CSC – IT Center for Science in Finland for computational resources. The work was funded by grants received from Emil Aaltonen foundation and Academy of Finland.

References

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6(9):1345–1382.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155.

Douglas Biber. 1988. *Variation across speech and writing*. Cambridge University Press, Cambridge.

Douglas Biber and Jesse Egbert. 2016. Using grammatical features for automatic register identification in an unrestricted corpus of documents from the open web. *Journal of Research Design and Statistics in Linguistics and Communication Science*, 2(1):3–36.

Douglas Biber and Jesse Egbert. 2019. *Register Variation Online*. Cambridge University Press, Cambridge.

Avrim L Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271.

Marina Bondi and Mike Scott. 2010. *Keyness in Texts*. John Benjamins Publishing Company.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Daria Dayter and Thomas Messerli. 2021. Persuasive language and features of formality on the rchangemyview subreddit. *Internet Pragmatics*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052.

Jesse Egbert and Doug Biber. 2019. Incorporating text dispersion into keyword analyses. *Corpora*, 14(1):77–104.

Jesse Egbert, Douglas Biber, and Mark Davies. 2015. Developing a bottom-up, user-based method of web register classification. *Journal of the Association for Information Science and Technology*, 66:1817–1831.

Costas Gabrielatos and Anna Marchi. 2011. Keyness: Matching metrics to definitions. In *Theoretical-methodological challenges in corpus approaches to discourse studies and some ways of addressing them*.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.

Kurt Hornik and Bettina Grün. 2014. movMF: an R package for fitting mixtures of von Mises-Fisher distributions. *Journal of Statistical Software*, 58(10):1–31.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Ron Kohavi and George H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint arXiv:2009.07896*.

Veronika Laippala, Jesse Egbert, Douglas Biber, and Aki-Juhani Kyröläinen. 2021. Exploring the role of lexis and grammar for the stable identification of register in an unrestricted corpus of web documents. *Language Resources and Evaluation*.

Veronika Laippala, Roosa Kyllönen, Jesse Egbert, Douglas Biber, and Sampo Pyysalo. 2019. Toward multilingual identification of online registers. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 292–297, Turku, Finland. Linköping University Electronic Press.

Gjorgji Madjarov, Vedrana Vidulin, Ivica Dimitrovski, and Dragi Kocev. 2019. Web genre classification with methods for structured output prediction. *Information Sciences*, 503:551 – 573.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *arXiv:1802.03426*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Philipp Petrenz and Bonnie Webber. 2011. Stable classification of text genres. *Computational Linguistics*, 37(2):385–393.

Martin A. Phillips. 1989. *Lexical structure of text*. English Language Research.

Punjaporn Pojanapunya and Richard Watson Todd. 2018. Log-likelihood and odds ratio: Keyness statistics for different purposes of keyword analysis. *Corpus Linguistics and Linguistic Theory*, 14(1):133–167.

Grusha Prasad, Yixin Nie, Mohit Bansal, Robin Jia, Douwe Kiela, and Adina Williams. 2021. To what extent do human explanations of model behavior align with actual model behavior? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 1–14, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liina Repo, Valtteri Skantsi, Samuel Rönnqvist, Saara Hellström, Miika Oinonen, Anna Salmela, Douglas Biber, Jesse Egbert, Sampo Pyysalo, and Veronika Laippala. 2021. Beyond the English web: Zero-shot cross-lingual and lightweight monolingual classification of registers. In *Proceedings of the EACL 2021 Student Research Workshop*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144.

Samuel Rönnqvist, Valtteri Skantsi, Miika Oinonen, and Veronika Laippala. 2021. Multilingual and zero-shot is closing in on monolingual web register classification. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 157–165, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Gideon Schwarz. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 2:461–464.

Mike Scott and Chris Tribble. 2006. *Textual patterns: Key words and corpus analysis in language education*. John Benjamins.

Serge Sharoff, Zhili Wu, and Katja Markert. 2010. The web library of babel: evaluating genre collections. In *Proceedings of LREC)*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*. ICLR.

Michael Stubbs. 2010. Three concepts of keywords. In Marina Bondi and Mark Scott, editors, *Keyness in texts: corpus linguistic investigations*, page 21–42. John Benjamins.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ashley Titak and Audrey Robertson. 2013. Dimensions of web registers: An exploratory multidimensional comparison. *Corpora*, 8:239–271.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76.

Raymond Williams. 1976. *Keywords: A vocabulary of culture and society*. Oxford University Press.

Appendix

Class	Docs	Tokens	Vocab.
Narrative	14,136	15,256k	498k
Informational description	7,460	10,171k	387k
Opinion	6,290	9,880k	360k
Interactive discussion	2,623	2,919k	151k
Informational persuasion	2,246	1,197k	93k
How-to	1,066	1,210k	78k
Lyrical	512	248k	26k
Spoken	470	961k	67k
Hybrids	4,545	5,939k	270k

Table 5: Quantitative descriptors of the data.

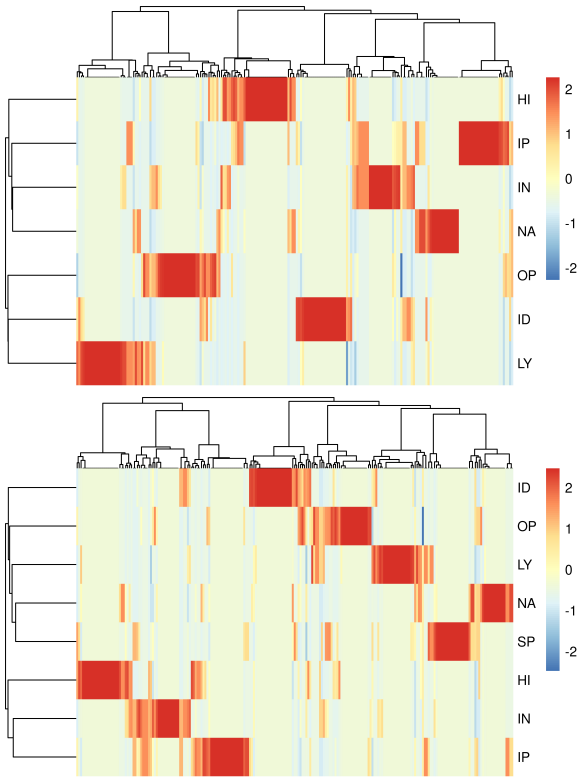

Figure 5: Cluster solution of the keywords relative to the classes extracted with SVMs (upper panel) and TF-IDF (lower panel). The row correspond to the classes and the columns to the keywords.

—— Keywords from SVMs ——							
How-to	**I. Discussion**	**I. Description**	**I. Persuasion**	**Lyrical**	**Narrative**	**Opinion**	**Spoken**
how	answers	abstract	description	lyrics	said	review	did
tips	resolved	or	book	comment	we	allah	aesthetic
add	forum	storyline	author	from	according	truly	we
step	quote	symptoms	brisbane	all	comments	and	applause
your	question	used	product	poem	says	relationship	very
recipe	thread	overview	dec	down	it	blog	interview
niche	chosen	summary	gift	me	last	jesus	true
dry	asker	courses	membership	poems	added	god	abc
tutorial	re	please	series	song	this	seems	there
mix	answer	causes	casino	oh	lovely	bible	that
use	etc	information	date	poetry	excited	ipod	think
pilates	dont	discusses	deals	gonna	confirmed	while	what
contract	originally	contact	pledge	yeah	announced	even	you
advance	posted	research	attracts	lord	they	character	hon
make	would	variety	pink	revolution	earlier	rather	do

—— Keywords from TF-IDF ——							
How-to	**I. Discussion**	**I. Description**	**I. Persuasion**	**Lyrical**	**Narrative**	**Opinion**	**Spoken**
using	question	information	book	lyrics	team	love	doing
add	etc	research	free	love	week	feel	kind
information	answer	number	author	song	game	let	music
start	try	using	amazon	chorus	against	doesn	feel
keep	someone	available	price	oh	says	god	love
tips	answers	including	read	http	government	read	yeah
try	bit	important	business	cause	season	fact	wanted
yourself	anything	based	order	www	told	money	bit
check	getting	must	love	baby	today	actually	working
important	problem	business	books	yeah	didn	book	started
page	doesn	health	information	feel	city	someone	didn
easy	feel	provide	add	gonna	man	man	actually
create	anyone	within	full	girl	night	doing	done
set	dont	often	product	wanna	second	ever	tell
list	keep	social	family	heart	news	business	play

Table 6: Top-15 keywords per class extracted by baseline methods.

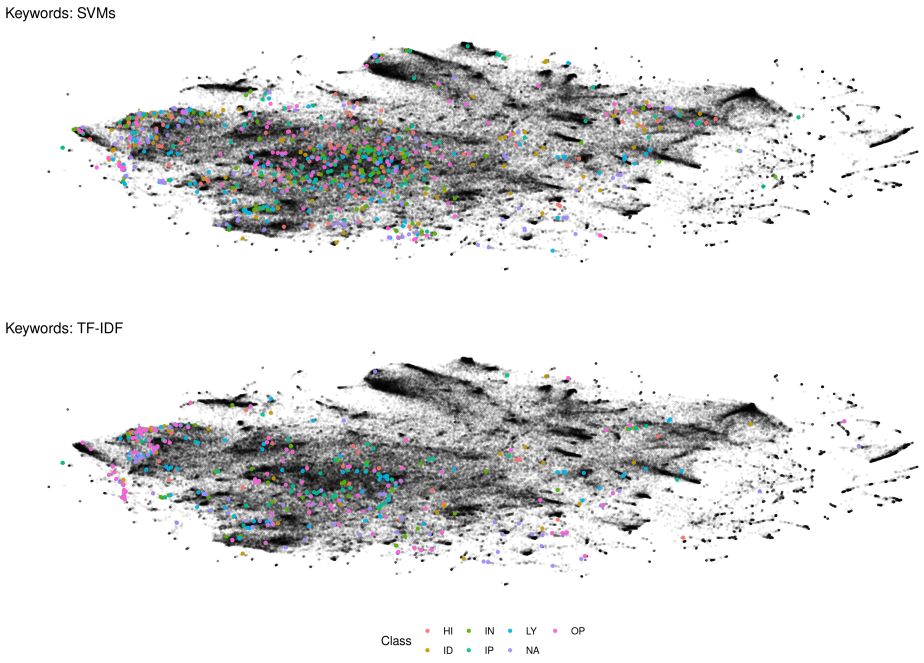

Figure 6: The lexical space of CORE and the keywords extracted with SVMs (upper panel) and TF-IDF (lower panel) are colored based on the class.