Comparing Apples to Apple: The Effects of Stemmers on Topic Models

Alexandra Schofield
Cornell University
Ithaca, NY 14853
xanda@cs.cornell.edu

David Mimno
Cornell University
Ithaca, NY 14853
mimno@cornell.edu

Abstract

Rule-based stemmers such as the Porter stemmer are frequently used to preprocess English corpora for topic modeling. In this work, we train and evaluate topic models on a variety of corpora using several different stemming algorithms. We examine several different quantitative measures of the resulting models, including likelihood, coherence, model stability, and entropy. Despite their frequent use in topic modeling, we find that stemmers produce no meaningful improvement in likelihood and coherence and in fact can degrade topic stability.

1 Introduction

Stemming is a popular way to reduce the size of a vocabulary in natural language tasks by conflating words with related meanings. Specifically, stemming aims to convert words with the same "stem" or root (e.g "creative" and "creator") to a single word type ("create"). Though originally developed in the context of information retrieval (IR) systems, stemmers are now commonly used as a preprocessing step in unsupervised machine learning tasks. It this work we consider one such application, topic modeling. Although stemmers are commonly used in topic models (Liu et al., 2010; Lo et al., 2015; Nan et al., 2015; Kamath S et al., 2015; Su, 2015; Jacobi et al., 2016), we find no empirical benefits for the practice.

One could conjecture several reasons to stem for semantic models. First, conflating semantically related words into one word type could improve model fit by intelligently reducing the space of possible models. Given that reducing the feature space randomly is already known to be potentially beneficial (Ganchev and Dredze, 2008), doing so in a semantically-inspired way might be even better. Second, stemmers could reduce the effect of small morphological differences on the stability of a learned model. Reducing the words "happy", "happily", and "happier" to one token may result in fewer possible models with divergent "happy" topics. Third, stemmers approximate intuitive word equivalence classes, so language models based on stemmed corpora inherit that semantic similarity, which may improve interpretability as perceived by human evaluators.

However, stemmers have the potential to be confusing, unreliable, and possibly even harmful in language models. First, many stemmers produce terms that are not recognizable English words and may be difficult to map back to a valid original word, such as "stai" as the Porter stem of "stay". Second, although stemming aids document retrieval for many languages, English is a notorious exception (Harman, 1991). In English, the complexity of compound affixes with meaning can lead to overstemming, such as "recondition," a word sharing a stem but not a root meaning with "recondite." These complexities can also lead to the incorrect conflation of words with the same root but divergent meaning such as "absolutely" and "absolution". Third, and most troubling, there are cases in which morphological variants of the same stem carry significantly different meanings. Conflating "apple" and "apples" is uncontroversial, but loses the distinction between a device manufacturer and a type of fruit.

Transactions of the Association for Computational Linguistics, vol. 4, pp. 287–300, 2016. Action Editor: Hal Daume III.
Submission batch: 2/2016; Published 7/2016.

Topic modeling is sensitive to preprocessing because of its dependence on a sparse vocabulary (Jockers and Mimno, 2013). In practice, however, preprocessing methods are typically neither detailed nor justified, leading to problems in reproducibility (Fokkens et al., 2013). We believe investigating the effects of stemming will inform researchers outside the core natural language processing community as to how to best preprocess their texts.

While stemmers are used in topic modeling, we know of no analysis focused on their effect. We draw inspiration from prior studies of the effects of stemming for other tasks and models (Harman, 1991; Han et al., 2012; Jivani, 2011; Rani et al., 2015) to apply rule-based stemmers to a variety of corpora to test their effect on topic models. We evaluate the quantitative fit of the models generated and the qualitative differences between differently-stemmed corpora to investigate the effects each stemmer has on a corpus. We hope that these results help guide future researchers as to how to select and evaluate stemmers for a given task and corpus.

2 Background

In this work we consider two categories of word normalization[1] methods: *rule-based stemmers*, or stemmers primarily reliant on rules converting one affix to another, and *context-based methods*, or strategies that use dictionaries and other contextual, inflectional, and derivational information to infer the correct word root Jivani (2011). We omit several language-independent strategies of text normalization, including those using Markov chains (Melucci and Orio, 2003) and clustering (Majumder et al., 2007). These methods are corpus-specific and error-prone, and we have not observed their use in topic modeling.

In our evaluation, we consider nine different methods of word normalization, given below with two-letter labels. In addition to including popular rule-based stemmers, we choose several simple stemmers that are stronger and weaker than the named stemmers, where *strength* refers to how much the vocabulary is reduced. We will sometimes use the more general term *conflation treatment* or simply *treatment* to refer to these methods with respect to our corpus. These are compared to the control, no-stemmer treatment, **NS**.

2.1 Rule-Based Treatments

The first category, *rule-based stemmers*, includes methods primarily governed by a set of rules that convert one affix to another. Most classic stemmers fit into this category, including the famous Porter stemmer. These methods are quick, but also limited: no concise rule set captures every English morphological exception, and these stemmers cannot use context to resolve ambiguous word types. They also are deterministic and consistent for each token: if word type A maps to stem B in one location, it will do so in every location word type A arises. Treatments of this type therefore are effectively equivalence relations over untreated words, with a *conflation class* being an equivalence class of word types under a conflation treatment t.

While Jivani (2011) refers to these as "truncation stemmers" or "affix removal stemmers," we find this naming confusing: stemmers rarely strictly truncate, and almost all stemmers aim to remove affixes. The core similarity of these methods is that all of the language-specific information used in these stemmers is encoded directly into the rules they apply.

Truncation Stemmers. k-truncation stemmers (Bhamidipati and Pal, 2007) remove all but the first k characters of a word. As a naïve, high-strength method, they serve as a good baseline for the relative effects of simple vocabulary reduction. We test four-truncation (**T4**) and five-truncation (**T5**). Five-truncation has strength close to a strong rule-based stemmer; levels below four are incoherent.

"S" Stemmer. The S-removal stemmer or "S" stemmer (**SS**) removes S-based endings using only three rules. Harman (1991) introduces the "S" stemming algorithm as a weaker and simpler counterpoint to more standard rule-based stemmers. As the rules are simple and good representatives of the types of rules employed by the other stemmers in this section, we include them in Table 1.

Lovins Stemmer. The Lovins stemmer (**LS**) is a rule-based stemmer using a two-step stemming algorithm (Lovins, 1968). These steps use long lists of rules, but the method is still fast and simple to imple-

[1] Other methods for word normalization include case folding and replacing classes of tokens with a constant (e.g. NUMBER for numerals).

If word ends with:	. . . and does not end with:	. . . replace ending with:
-ies	-aies, -eies	-y
-es	-aes, -ees, -oes	-e
-s	-ss, -us	-

Table 1: The "S" stemmer of Harman (1991) consists of three simple rules in order. Only the first rule applicable in the first column is applied.

ment and is generally considered a strong stemmer.

Porter and Porter2 Stemmers. The Porter stemmer (Porter, 1980), one of the most popular in current use, is a slightly less strong and more intricate stemmer than Lovins'. It uses five phases of rules and conditions that match patterns of vowel and consonant sequences. Porter later created a slightly improved version of the Porter stemmer for Snowball, a programming language for rule-based stemmers (Porter, 2001). We use both the original stemmer (**P1**) and the new version (**P2**) in our evaluation.

Paice/Husk Stemmer. The Paice/Husk stemmer (**PH**), or Lancaster stemmer, iterates indefinitely over the same rule list, with some rules only applying to unmodified words and others terminating iteration (Paice, 1990). While slightly more complicated in rule structure, the Paice/Husk stemmer is similar to the Lovins stemmer in strength.

2.2 Context-Based Treatments

While the methods above are fast, they are imprecise, as a limited set of rules cannot account for all possible morphological exceptions. Subtleties such as the difference between "frosting" windows and cake "frosting" are lost without contextual information. The methods below use tools such as dictionaries, inflectional analysis, and part-of-speech inference to determine the correct conflated form of a word. As such, they may not consistently reduce the same word type to the same form. However, these tools also demand more computational resources; for our data, lemmatizing the corpus took more computational time than training the topic model.

Krovetz Stemmer. The Krovetz stemmer (Krovetz, 1993) uses inflectional analysis and a dictionary to determine correct forms of words before removing word endings. This process is complex, but the stemmer itself is weak, as it aims less at conflating words with different parts of speech than normalizing verb forms and removing pluralization. The dictionary itself is crucial for implementation; for our Krovetz stemmer treatment (**KS**), we use the Lemur Project implementation.

Lemmatizer. Lemmatizers use a database of lemmas, or standardized word forms, in order to find the best normalized word form for a given token. While the method is orders of magnitude slower than rule-based stemmers, it is also much more principled and extremely unlikely to over-conflate. We use the WordNet-based lemmatizer (**WL**) implemented in the Natural Language ToolKit (Bird et al., 2009) along with a Stanford POS Tagger (Toutanova et al., 2003) on the unmodified text to provide auxiliary part-of-speech information for the lemmatizer.

3 Model and Data

In this paper, we focus on modeling topics in English datasets using Latent Dirichlet Allocation (LDA), a generative model for documents based upon their topics (Blei et al., 2003). A topic ϕ in this context is a multinomial probability distribution over words, without any embedded semantic model of how the words are connected. In LDA, each document has a multinomial distribution θ over topics; a document d is generated by choosing a number of words, and for each word first sampling a topic k from θ_d, then a word w from the distribution over words ϕ_k associated with topic k.

The name Latent Dirichlet Allocation comes from the assumptions that information about each word's topic or the original distributions θ and ϕ is latent (i.e. unobserved) and that the topic and word distributions θ and ϕ are drawn from Dirichlet distributions: $\theta \sim \text{Dir}(\alpha)$, and $\phi \sim \text{Dir}(\beta)$. Using this model, one can attempt to infer the most likely topics to generate a corpus for some preset number of topics K. However, the optimization problem is non-convex and intractable to solve analytically, and is thus generally solved using iterative techniques such as Gibbs sampling, expectation maximization, or variational inference. The resulting topics frequently display themes within the common words in a topic that can be used for classification, search, and recommendation systems.

Because stemming affects the vocabulary distribution of a corpus, the optimal parameters of topic model inference will vary depending on treatment. We us adaptive optimization of both Dirichlet hyperparameters α and β. We use an asymmetric α and symmetric β to obtain the best model fit in accordance with Wallach et al. (2009).

In order to test the various word normalization treatments, we used an existing Python library for the Lovins, Paice/Husk, and both Porter algorithms (Chaput, 2010), modified to correct errors in implementation. We implemented our own truncation stemmers and S-removal stemmer. We applied each stemmer to each word token in four corpora: articles from ArXiv in early 2015,[2] articles from The New York Times in 2007 (Sandhaus, 2008), biographies from IMDb,[3] and reviews from the Yelp Dataset Challenge.[4] Corpora were partitioned into 75% training documents, 25% test documents and lower-cased before conflation, which was performed per-sentence on lower-cased text. After treatment, we remove stopwords, digits, and punctuation. Table 2 shows details of the corpora, and Table 3 shows examples of each treatment.[5] We train topic models using MALLET (McCallum, 2002) for K = 10, 50, and 200, with at least nine models for each corpus, treatment, and K combination.

	Training Data		Evaluation Data	
Corpus	# docs	# toks	# docs	# toks
ArXiv articles	17.1K	58.4M	5.7 K	19.5M
IMDb bios	84.6K	9.13M	28.2K	3.05M
NYT articles	29.4K	8.81M	9.79K	2.98M
Yelp reviews	844K	43.1M	281K	14.4M

Table 2: Training and test corpora represent considerable variance in content, size of corpus, average length of document, and proportion of training to test data.

4 Evaluations

In order to evaluate the differences between conflation treatments of these corpora, we want to look at a variety of different types of evaluation of topic models. Unfortunately, as described later, standard evaluations of topic quality such as held-out likelihood and coherence are implicitly affected by the size of the vocabulary. To be able to compare different treatments without simply favoring the maximum possible vocabulary reduction, we create modified versions of several existing classic evaluations as well as new metrics for understanding differences in models at the level of word types instead of topics.

4.1 Held-Out Likelihood

Strong stemmers can improve the joint probability of documents occurring without improving the quality of the model. As we reduce the size of the vocabulary, each topic-word distribution is spread over fewer possible words; at its extreme, the probability of any corpus under a zero-truncation stemmer would be 1.0. Experiments confirmed that for these treatments, the standard held-out likelihood score $\mathcal{L}$ of the test corpus based on the trained model ordered stemmers by how much they reduce the vocabulary, assigning the highest likelihood to those treatments with the smallest vocabularies.

To account for the likelihood improvement caused by reducing vocabulary size, we normalize a model with K topics by the likelihood of a smoothed unigram language model with the same β parameter. We calculate from the normalized log likelihood $\mathcal{L}_{\text{norm}}$ a per-token metric PTLL$_{\text{norm}}$ to put corpora of different lengths on a comparable scale. We compute the unigram model probability as a smoothed multinomial with prior β, number of instances of word type w in a corpus n_w, vocabulary size W and total token count N:

$$\mathcal{L}_{\text{unigram}} = \prod_j \prod_i \frac{n_{w_{ij}} + \beta}{N + W\beta} \tag{1}$$

$$\mathcal{L}_{\text{norm}} = \mathcal{L} / \mathcal{L}_{\text{unigram}} \tag{2}$$

$$\text{PTLL}_{\text{norm}} = \frac{\log(\mathcal{L}_{\text{norm}})}{N} = \frac{\log \mathcal{L}}{N} - \frac{log(\mathcal{L}_{\text{unigram}})}{N}. \tag{3}$$

Our resulting metric measures how much on average the introduction of multiple topics improves the probability of each token occurring.

4.2 Topic Coherence

Though log likelihood describes the statistical likelihood of the topic model generating the corpus, it

[2] Retrieved from ArXiv (`http://www.arxiv.org`).

[3] Courtesy of IMDb (`http://www.imdb.com`).

[4] Retrieved from Yelp (`http://www.yelp.com/dataset_challenge`).

[5] Our code can be found at `https://github.com/heraldicsandfox/stemmers`.

Original	This location does not have good service. Went through drive-through and they forgot our drinks and our sides. While they were preparing what they forgot, we could see another girl who had her back to us and it was obvious that she was on her phone. Any other KFC would be better.
Tokenized	this location does not have good service went through drive through and they forgot our drinks and our sides while they were preparing what they forgot we could see another girl who had her back to us and it was obvious that she was on her phone any other kfc would be better
Stopped	location good service drive forgot drinks sides preparing forgot girl back obvious phone kfc

NS	location	good	service	drive	forgot	drinks	sides	preparing	forgot	girl	back	obvious	...
T4	loca	good	serv	driv	forg	drin	side	prep	forg	girl	back	obvi	...
T5	locat	good	servi	drive	forgo	drink	sides	prepa	forgo	girl	back	obvio	...
LO	loc	good	servic	dr	forgot	drink	sid	prepar	forgot	girl	back	obv	...
P1	locat	good	servic	drive	forgot	drink	side	prepar	forgot	girl	back	obviou	...
P2	locat	good	servic	drive	forgot	drink	side	prepar	forgot	girl	back	obvious	...
PH	loc	good	serv	driv	forgot	drink	sid	prep	forgot	girl	back	obvy	...
SS	location	good	service	drive	forgot	drink	side	preparing	forgot	girl	back	obvious	...
KR	location	good	service	drive	forgot	drink	side	prepare	forgot	girl	back	obvious	...
WL	location	good	service	drive	forget	drink	side	prepare	forget	girl	back	obvious	...

Table 3: A demonstration of the steps of preprocessing on a Yelp review.

does not necessarily indicate topics that are semantically coherent to a human observer. To measure this, we use the topic coherence measure proposed by Mimno et al. (2011). This metric is defined for a given topic k and a list of the top M words of a topic $v_1^k, \dots v_M^k$ as

$$C(k) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_l, v_m) + \beta}{D(v_l) + \beta} \tag{4}$$

where $D(v_l)$ is the number of documents in which word v_l occurs and $D(v_l, v_m)$ is the number of documents in which both words v_l and v_m occur. This metric is similar to pointwise mutual information Lau et al. (2014), but instead of using a sliding window over the text to determine co-occurrence, it uses full documents as discrete windows.

To avoid biasing towards the smaller vocabularies of stemmed datasets, we use the token-topic assignments output by the topic model with the list of untreated tokens to produce untreated top keywords for each topic. We then use the original untreated corpus and these new keywords to compute coherence values. This allows us to observe whether conflation treatments map tokens to the same topic in a more coherent way than untreated corpora would. We experimented with using Wikipedia as a reference corpus, but found it too general a reference for a semantic model in a narrow context such as a scientific paper or an actor biography.

4.3 Clustering Consistency

If we treat topic models as clusterings of tokens, we can evaluate how consistent those clusters are. Variation of information (VOI), a symmetric measurement of difference between clusterings, allows us to evaluate how much of a difference stemming makes in the topics formed (Meilă, 2003; Grimmer and King, 2011). Although some degree of variation is inevitable between different trials with the same treatment due to randomness in the inference algorithm, stemming may affect how much occurs. We use two VOI-based metrics to examine treatment stability and differences: intra-treatment VOI and inter-treatment VOI. Intra-treatment VOI is VOI between models trained with different random initializations but the same treatment. Correspondingly, inter-treatment VOI is the VOI between outputted topic assignments from different treatments. If the inter-treatment VOI is equal to the VOI between trials of the same treatment, we infer that the change in treatment has made a negligible difference in the assignment of tokens to topics.

4.4 Influential Words

The metrics above are all summary statistics that measure different types of overall topic model quality. However, to understand why these metrics are affected the way they are, we also need some way to examine the individual components we have affected: the word types available in our documents.

We use two heuristics to identify words that are most affected by a given treatment. The first uses inferred token probabilities in the test corpus. We want a scoring function of untreated word types that is positive if the estimated joint probability of tokens of a particular pre-treatment type increases after treatment, and negative if it decreases. We also want the magnitude of the score to correspond with both the difference in probability across all tokens and the relative informativeness of that token in distinguishing documents or topics.

For a given word type w from the untreated corpus and function t applying some conflation treatment, we compute the word type probability, TP_{wt}, as

$$\sum_{d=1}^{D}\sum_{i=1}^{N_d} I[x_{di} = w] \log(P(t(x_{di}) = t(w)|\ldots)), \tag{5}$$

where D is the number of documents, N_d is the number of tokens in document d, x_{di} is the untreated word type of token i in document d and $t(x_{di})$ is the treated type, $I[x_{di} = w]$ is the indicator function that is 1 if $x_di = w$ and zero otherwise, and $P(t(x_{di}) = t(w)|\ldots)$ is shorthand for the held-out likelihood estimate of treated token $t(x_{di})$ having the type w generated using the left-to-right technique given the inferred parameters θ, ϕ and hyperparameters α, β of the trained model.

We average the quantity in Equation 5 across all topic models of the same corpus, topic count, and treatment to get $\overline{TP}_{wt}$. In order to compute a relative score of the amount of probability improvement of an individual treatment for a word type from the no-stemmer treatment t_0, we take the difference between topic probabilities, weighted by inverse document frequency (idf) to favor words that are specific to particular documents. Our final score function is

$$TPscore_{wt} = (\overline{TP}_{wt} - \overline{TP}_{wt_0}) \log\left(\frac{D}{D_w}\right), \tag{6}$$

where D_w is the number of documents of the total D containing at least one token of type w. The lowest negative scores indicate higher probability and importance of the unstemmed form of the token, while high positive scores indicate higher probability and importance of the stemmed form. While this does not produce a symmetric distribution, as we have not accounted for the increased probability of each word in a smaller vocabulary, it allows us to sort words by how much their probability of occurring has changed between treatments and how much that word affects the corpus as a whole.

The second heuristic tests whether stemming increases or decreases certainty of the topic assignment for each stemmed word type. Intuitively, correct conflation should reduce the information entropy across tokens of a given conflation class by forcing words with the same root to be treated as a single word in inference. Topic models are intuitively better when words are sparsely distributed across topics; consequently, we prefer lower entropy across topics, or mass concentrated in only a few topics. A negative value for a word type under this entropy metric favors the stemmed corpus, while a positive score favors the untreated corpus.

In this case, for a given word type w, we use the topic assignments from the final iteration of Gibbs sampling to compute the number of instances of w assigned to each topic k. To preserve the sparsity inferred by the algorithm, we use this to generate a maximum-likelihood estimate of the probability distribution of w being assigned to each topic, from which we can compute the Shannon entropy:

$$H_{wt}(k) = -\sum_{k=1}^{K} \frac{N_{wk}}{N_w} \log\left(\frac{N_{wk}}{N_w}\right), \tag{7}$$

where N_{wk} is the count of all tokens of type w assigned to topic k. For each treated form of a word w by a treatment t, we also consider the inverse image $t^{-1}(w)$, or the set of all words that stem to have form w. We therefore compute a change in entropy using average $\bar{H}_{wt}$ across all trials with treatment t and control t_0 for a given corpus and topic count,

$$\Delta H_{wt}(k) = \bar{H}_{wt}(k) - \bar{H}_{t^{-1}(w)t_0}(k), \tag{8}$$

where $\bar{H}_{t^{-1}(w)t_0}$ is the information entropy for the topic-word counts summed across all untreated types that conflate to type w under treatment t.

5 Results

To evaluate the effects of conflation on topic models, we produced several thousand inferred models. We apply the metrics described in Section 4, computing means and standard errors across trials with the same topic count, corpus, and treatment where possible to ensure significance.

5.1 Treatment Strength

Many factors contribute to the general concept of “strength” of a stemmer, but the most obvious signal is the amount by which a stemmer reduces the vocabulary of a corpus. After stopword removal, we count the total number of unique word types in our stemmed corpus for each treatment and training corpus, as well as the average number of characters in each word after treatment. Comparing type-token ratios of rule-based stemmers to the untreated corpus gives a measurement of the ratio of untreated words to conflation classes under that treatment. We display these counts in Figure 1.

The results of these stemming treatments already demonstrate that stemmer strength can depend heavily on the type of corpus on which it is applied. For instance, the Krovetz stemmer actually increases the size of the vocabulary of ArXiv, whereas it produces more vocabulary reduction than the lemmatizer on both IMDb and Yelp. The proportions of rule-based stemmer type-token ratios are consistent across corpora, with the exception of truncation on arXiv. The frequent use of scientific prefixes (such as “inter” and “anti”) and bad conversion from PDF format in arXiv lead truncation stemmers to conflate at a higher rate than they do on other corpora with respect to other rule-based stemmers. The three different light stemming methods — the S-stemmer, the Krovetz stemmer, and the WordNet lemmatizer — perform similarly on the IMDb corpus, but vary substantially across the other three corpora.

Character-token ratios vary less between corpora than type-token ratios. Five-truncation produces words with an average length near the Paice-Husk and Lovins stemmers. Not surprisingly, S-stemming produces an average word length slightly less than the untreated corpus, while the Krovetz stemmer and WordNet lemmatizer vary in strength across corpora.

We also verify some expected results for these stemmers: truncation stemmers are very strong, with four-truncation reducing vocabulary size to one-fourth or one-fifth of the original. The Porter stemmers behave similarly to each other, with slightly more liberal stemming by the Porter2 stemmer on all corpora but ArXiv. The Paice-Husk and Lovins stemmers are both stronger than Porter, while the S-stemmer is consistently weaker. While the vocabulary of a corpus affects the strength of each stemmer, it does little to affect the strengths of the rule-based stemmers relative to each other.

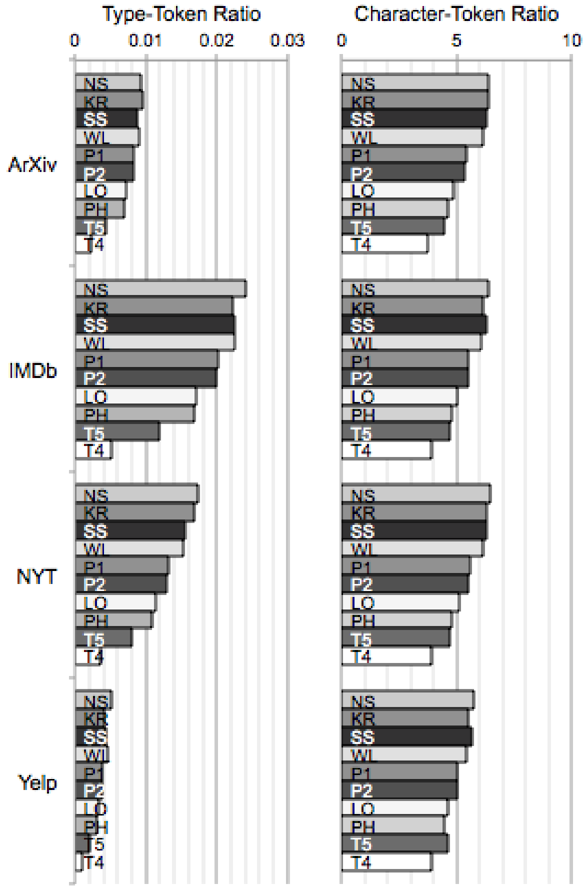


Figure 1: Type-token ratio and character-token ratio vary substantially across training corpora and conflation treatments. Due to the context-sensitive stemming done by the Krovetz stemmer, it is possible for one untreated word type to map to multiple stemmed types, producing a greater type-to-token ratio for the ArXiv version of the Krovetz stemmer than for the original untreated corpus.

5.2 Held-Out Likelihood

Using our normalized log likelihood measure from Equation 3, we can compare likelihoods across all our different treatments, as shown in Figure 2. We observe for all standard rule-based stemmers treatments provide little likelihood benefit apart from reducing the vocabulary size; the Porter stemming

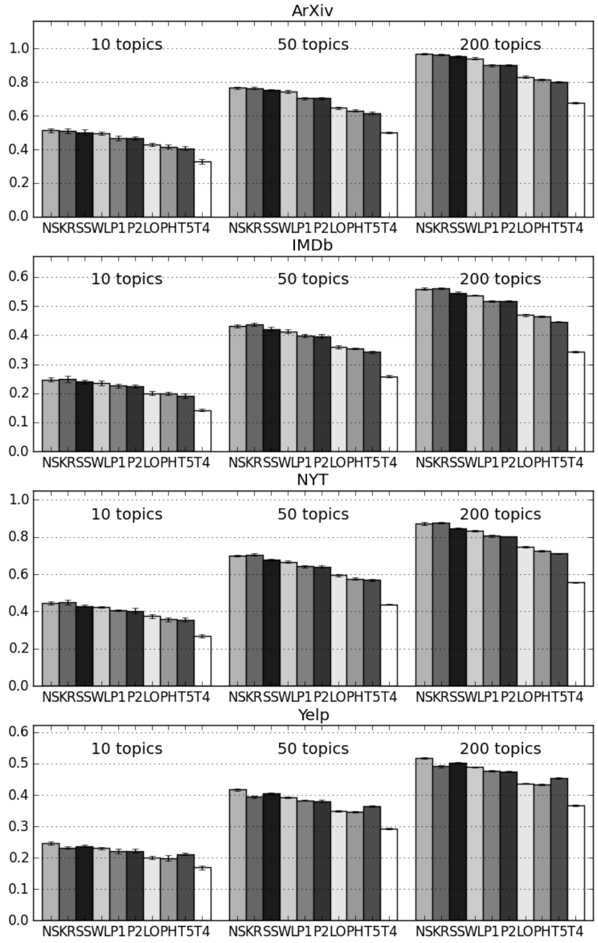


Figure 2: While light conflation treatments may help particular corpora, word conflation generally decreases the statistical fit of a topic model proportionally to its strength as measured in normalized log likelihood. Confidence intervals are the $p = 0.99$ range of belonging to the distribution of that treatment's normalized log likelihoods across at least 9 samples each. Higher values of normalized log likelihood represent better model fit.

treatments result in normalized log likelihoods significantly lower than the unstemmed treatment. Statistically, the Porter stemmers do not appear to be improving the quality of the model; they are merely reducing the possible unigrams it could generate in a moderately principled way. Both Paice/Husk and Lovins have the same problem, but as they are stronger stemmers, problems of overconflation seem to reduce the quality further.

More surprising, however, is the mediocre performance of the WordNet lemmatizer. The fact that Yelp and IMDb do not see an improvement with use of the lemmatizer is easy to explain away: these corpora contain slang, misspellings, and plenty of proper names, enough to make lemmatization a challenge. However, we see the same result in the case of New York Times articles, an ideal corpus for topic modeling. While there are still many named entities, they arise in carefully-edited text with standardized journalistic vocabulary.

Other observations are less surprising. Five-truncation produces likelihoods comparable to the stronger Lovins and Paice/Husk stemmers, and significantly better than either for the 50-topic Yelp model. This may relate to the irregularities of review text: words elongated for emphasis (e.g. "helloooo") and other oddities of online informal English are hard for rule-based suffix stemmers to handle but still benefit from naïve forms of conflation. The Porter and Porter2 stemmers are not significantly different in any case, which serves as comforting validation that those not using the new generation of the Porter stemmer are not losing much.

5.3 Topic Coherence

Log likelihood measures can tell us about statistical fit, but do not necessarily tell us about the actual apparent coherence of the model in terms of conceptual similarity of words in a topic. In Figure 3, we display the negative average coherence scores from Equation 4 for each treatment. The hypothesis we test is that using a conflation treatment should map morphologically different words with a shared concept to the same word, automatically constraining the topic model to ensure closely-related words are proportionally present in the same topics.

Our results do not conform to this intuition. The majority of treatments are statistically indistinguishable from the untreated control with respect to coherence. The relative effects of these treatments on coherence are magnified as the number of topics increases; while no ArXiv treatment differs significantly in coherence at 10 topics, at 200, the four strongest treatments (Lovins, Paice-Husk, five-truncation and four-truncation) are significantly worse. Four-truncation suffers a similar effect on IMDb at 50 and 200 topics. In contrast, four-truncation actually improves in coherence compared to other treatments on Yelp as the number of topics increases, reaching a significant level at 200 topics.

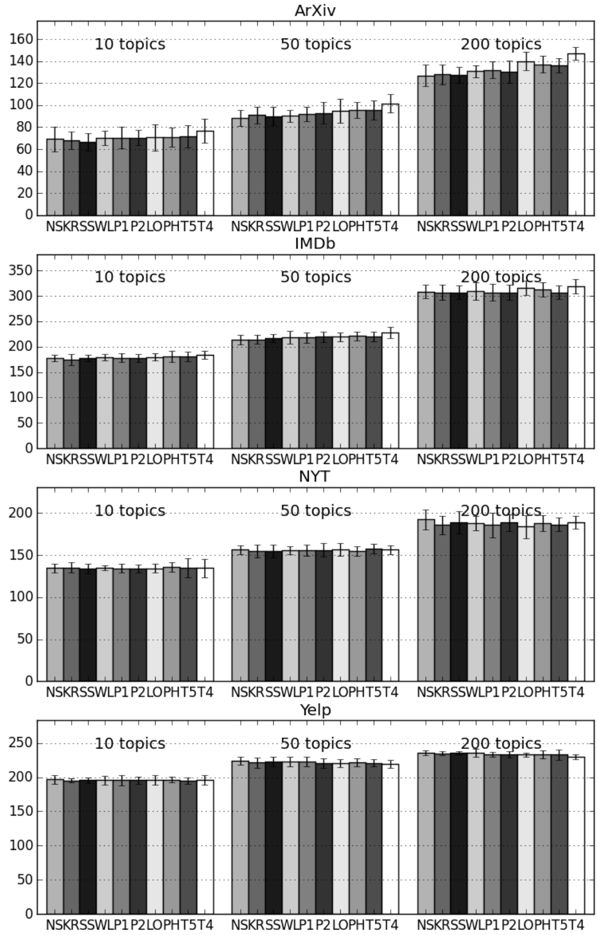


Figure 3: Conflation treatments introduce no significant difference in almost all cases in the resulting average negative topic coherence of each model according to token assignments. Smaller values indicating better coherence, and error bars represent the $p = 0.99$ range of possible mean values.

Given the lack of substantial statistical difference across a variety of treatments, it seems safe to conclude that the use of stemmers is not substantially improving the encoding of word similarities in these topics. The topic model itself on the untreated corpus is perhaps already doing as good a job ensuring that words in the same conflation class have statistically similar topic distributions.

Unstemmed topics sometimes contain words from the same conflation class (e.g. "restaurant" versus "restaurants"). While these might give a slight advantage in coherence measures, this case implies that the stemmers are not necessary for bringing together morphological variants in topics.

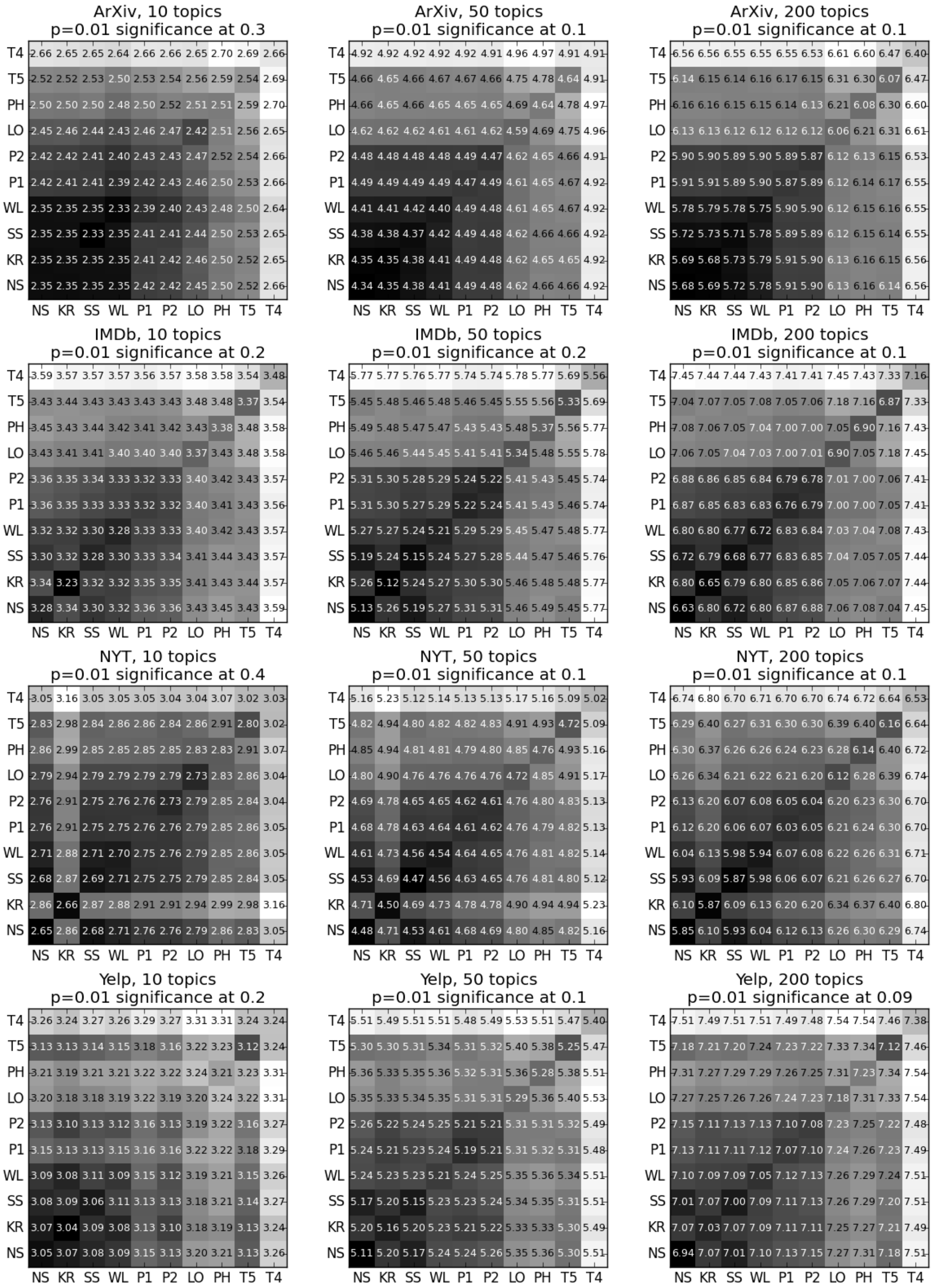
5.4 Clustering Consistency

Another hypothesized effect of stemming is that it will produce more consistent results by reducing the sensitivity of related words to random initialization. We can use variation of information (VOI) to understand how these models differ from each other relative to how much they vary between random initializations. We summarize the results in Figure 4.

Within statistical error bounds, intra-treatment VOI is always less than or equal to the variation across treatments, and VOI increases as the number of topics increases. On ArXiv, the light treatments — the Krovetz stemmer, S-stemmer, and WordNet lemmatizer — behave indistinguishably from the untreated corpus. The intra-treatment VOI trend shows that stronger treatments generally result in less consistent models. This contradicts the intuition that stemming will help place words with similar meaning into the same topic. While stemming constrains all conflated word types to share one probability in each topic, it does not ensure that those probability distributions will favor few topics.

There are two striking exceptions to this trend. The first is the Krovetz stemmer. The intra-treatment VOI of Krovetz topic models stays closer to that of the untreated corpus than the S-stemmer or the WordNet lemmatizer. However, the higher inter-treatment VOI between Krovetz and the unstemmed corpus suggests that the Krovetz stemmer produces small but significant changes in the optima of the topic model. For IMDb, NYT, and Yelp at 200 topics, and NYT again at 50 topics, the VOI between the untreated corpus and Krovetz-stemmed corpus is significantly greater than the VOI of the untreated corpus with itself. In contrast, the variation of information between untreated and S-stemmed corpora is only negligibly higher than the intra-treatment VOI of the S-stemmer. This is interesting given the reputation of Krovetz as a weak stemmer.

The second exception is the five-truncation stemmer. Though a very strong stemmer, its VOI is indistinguishable from the heavier Lovins and Paice-Husk stemmers on most corpora, but when applied to Yelp with 200 topics, it actually does significantly better than either, in both intra-treatment and inter-treatment VOI with the unstemmed corpus. This effect can be seen to a less significant extent in mod-



ArXiv, 10 topics
p=0.01 significance at 0.3

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	2.66	2.65	2.65	2.64	2.66	2.66	2.65	2.70	2.69	2.66
T5	2.52	2.52	2.53	2.50	2.53	2.54	2.56	2.59	2.54	2.69
PH	2.50	2.50	2.50	2.48	2.50	2.52	2.51	2.51	2.59	2.70
LO	2.45	2.46	2.44	2.43	2.46	2.47	2.42	2.51	2.56	2.65
P2	2.42	2.42	2.41	2.40	2.43	2.43	2.47	2.52	2.54	2.66
P1	2.42	2.41	2.41	2.39	2.42	2.43	2.46	2.50	2.53	2.66
WL	2.35	2.35	2.35	2.33	2.39	2.40	2.43	2.48	2.50	2.64
SS	2.35	2.35	2.33	2.35	2.41	2.41	2.44	2.50	2.53	2.65
KR	2.35	2.35	2.35	2.35	2.41	2.42	2.46	2.50	2.52	2.65
NS	2.35	2.35	2.35	2.35	2.42	2.42	2.45	2.50	2.52	2.66

ArXiv, 50 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	4.92	4.92	4.92	4.92	4.92	4.91	4.96	4.97	4.91	4.91
T5	4.66	4.65	4.66	4.67	4.67	4.66	4.75	4.78	4.64	4.91
PH	4.66	4.65	4.66	4.65	4.65	4.65	4.69	4.64	4.78	4.97
LO	4.62	4.62	4.62	4.61	4.61	4.62	4.59	4.69	4.75	4.96
P2	4.48	4.48	4.48	4.48	4.49	4.47	4.62	4.65	4.66	4.91
P1	4.49	4.49	4.49	4.49	4.47	4.49	4.61	4.65	4.67	4.92
WL	4.41	4.41	4.42	4.40	4.49	4.48	4.61	4.65	4.67	4.92
SS	4.38	4.38	4.37	4.42	4.49	4.48	4.62	4.66	4.66	4.92
KR	4.35	4.35	4.38	4.41	4.49	4.48	4.62	4.65	4.65	4.92
NS	4.34	4.35	4.38	4.41	4.49	4.48	4.62	4.66	4.66	4.92

ArXiv, 200 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	6.56	6.56	6.55	6.55	6.55	6.53	6.61	6.60	6.47	6.40
T5	6.14	6.15	6.14	6.16	6.17	6.15	6.31	6.30	6.07	6.47
PH	6.16	6.16	6.15	6.15	6.14	6.13	6.21	6.08	6.30	6.60
LO	6.13	6.13	6.12	6.12	6.12	6.12	6.06	6.21	6.31	6.61
P2	5.90	5.90	5.89	5.90	5.89	5.87	6.12	6.13	6.15	6.53
P1	5.91	5.91	5.89	5.90	5.87	5.89	6.12	6.14	6.17	6.55
WL	5.78	5.79	5.78	5.75	5.90	5.90	6.12	6.15	6.16	6.55
SS	5.72	5.73	5.71	5.78	5.89	5.89	6.12	6.15	6.14	6.55
KR	5.69	5.68	5.73	5.79	5.91	5.90	6.13	6.16	6.15	6.56
NS	5.68	5.69	5.72	5.78	5.91	5.90	6.13	6.16	6.14	6.56

IMDb, 10 topics
p=0.01 significance at 0.2

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	3.59	3.57	3.57	3.57	3.56	3.57	3.58	3.58	3.54	3.48
T5	3.43	3.44	3.43	3.43	3.43	3.43	3.48	3.48	3.37	3.54
PH	3.45	3.43	3.44	3.42	3.41	3.42	3.43	3.38	3.48	3.58
LO	3.43	3.41	3.41	3.40	3.40	3.40	3.37	3.43	3.48	3.58
P2	3.36	3.35	3.34	3.33	3.32	3.33	3.40	3.42	3.43	3.57
P1	3.36	3.35	3.33	3.33	3.32	3.32	3.40	3.41	3.43	3.56
WL	3.32	3.32	3.30	3.28	3.33	3.33	3.40	3.42	3.43	3.57
SS	3.30	3.32	3.28	3.30	3.33	3.34	3.41	3.44	3.43	3.57
KR	3.34	3.23	3.32	3.32	3.35	3.35	3.41	3.43	3.44	3.57
NS	3.28	3.34	3.30	3.32	3.36	3.36	3.43	3.45	3.43	3.59

IMDb, 50 topics
p=0.01 significance at 0.2

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	5.77	5.77	5.76	5.77	5.74	5.74	5.78	5.77	5.69	5.56
T5	5.45	5.48	5.46	5.48	5.46	5.45	5.55	5.56	5.33	5.69
PH	5.49	5.48	5.47	5.47	5.43	5.43	5.48	5.37	5.56	5.77
LO	5.46	5.46	5.44	5.45	5.41	5.41	5.34	5.48	5.55	5.78
P2	5.31	5.30	5.28	5.29	5.24	5.22	5.41	5.43	5.45	5.74
P1	5.31	5.30	5.27	5.29	5.22	5.24	5.41	5.43	5.46	5.74
WL	5.27	5.27	5.24	5.21	5.29	5.29	5.45	5.47	5.48	5.77
SS	5.19	5.24	5.15	5.24	5.27	5.28	5.44	5.47	5.46	5.76
KR	5.26	5.12	5.24	5.27	5.30	5.30	5.46	5.48	5.48	5.77
NS	5.13	5.26	5.19	5.27	5.31	5.31	5.46	5.49	5.45	5.77

IMDb, 200 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	7.45	7.44	7.44	7.43	7.41	7.41	7.45	7.43	7.33	7.16
T5	7.04	7.07	7.05	7.08	7.05	7.06	7.18	7.16	6.87	7.33
PH	7.08	7.06	7.05	7.04	7.00	7.00	7.05	6.90	7.16	7.43
LO	7.06	7.05	7.04	7.03	7.00	7.01	6.90	7.05	7.18	7.45
P2	6.88	6.86	6.85	6.84	6.79	6.78	7.01	7.00	7.06	7.41
P1	6.87	6.85	6.83	6.83	6.76	6.79	7.00	7.00	7.05	7.41
WL	6.80	6.80	6.77	6.72	6.83	6.84	7.03	7.04	7.08	7.43
SS	6.72	6.79	6.68	6.77	6.83	6.85	7.04	7.05	7.05	7.44
KR	6.80	6.65	6.79	6.80	6.85	6.86	7.05	7.06	7.07	7.44
NS	6.63	6.80	6.72	6.80	6.87	6.88	7.06	7.08	7.04	7.45

NYT, 10 topics
p=0.01 significance at 0.4

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	3.05	3.16	3.05	3.05	3.05	3.04	3.04	3.07	3.02	3.03
T5	2.83	2.98	2.84	2.86	2.86	2.84	2.86	2.91	2.80	3.02
PH	2.86	2.99	2.85	2.85	2.85	2.85	2.83	2.83	2.91	3.07
LO	2.79	2.94	2.79	2.79	2.79	2.79	2.73	2.83	2.86	3.04
P2	2.76	2.91	2.75	2.76	2.76	2.73	2.79	2.85	2.84	3.04
P1	2.76	2.91	2.75	2.75	2.76	2.76	2.79	2.85	2.86	3.05
WL	2.71	2.88	2.71	2.70	2.75	2.76	2.79	2.85	2.86	3.05
SS	2.68	2.87	2.69	2.71	2.75	2.75	2.79	2.85	2.84	3.05
KR	2.86	2.66	2.87	2.88	2.91	2.91	2.94	2.99	2.98	3.16
NS	2.65	2.86	2.68	2.71	2.76	2.76	2.79	2.86	2.83	3.05

NYT, 50 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	5.16	5.23	5.12	5.14	5.13	5.13	5.17	5.16	5.09	5.02
T5	4.82	4.94	4.80	4.82	4.82	4.83	4.91	4.93	4.72	5.09
PH	4.85	4.94	4.81	4.81	4.79	4.80	4.85	4.76	4.93	5.16
LO	4.80	4.90	4.76	4.76	4.76	4.76	4.72	4.85	4.91	5.17
P2	4.69	4.78	4.65	4.65	4.62	4.61	4.76	4.80	4.83	5.13
P1	4.68	4.78	4.63	4.64	4.61	4.62	4.76	4.79	4.82	5.13
WL	4.61	4.73	4.56	4.54	4.64	4.65	4.76	4.81	4.82	5.14
SS	4.53	4.69	4.47	4.56	4.63	4.65	4.76	4.81	4.80	5.12
KR	4.71	4.50	4.69	4.73	4.78	4.78	4.90	4.94	4.94	5.23
NS	4.48	4.71	4.53	4.61	4.68	4.69	4.80	4.85	4.82	5.16

NYT, 200 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	6.74	6.80	6.70	6.71	6.70	6.70	6.74	6.72	6.64	6.53
T5	6.29	6.40	6.27	6.31	6.30	6.30	6.39	6.40	6.16	6.64
PH	6.30	6.37	6.26	6.26	6.24	6.23	6.28	6.14	6.40	6.72
LO	6.26	6.34	6.21	6.22	6.21	6.20	6.12	6.28	6.39	6.74
P2	6.13	6.20	6.07	6.08	6.05	6.04	6.20	6.23	6.30	6.70
P1	6.12	6.20	6.06	6.07	6.03	6.05	6.21	6.24	6.30	6.70
WL	6.04	6.13	5.98	5.94	6.07	6.08	6.22	6.26	6.31	6.71
SS	5.93	6.09	5.87	5.98	6.06	6.07	6.21	6.26	6.27	6.70
KR	6.10	5.87	6.09	6.13	6.20	6.20	6.34	6.37	6.40	6.80
NS	5.85	6.10	5.93	6.04	6.12	6.13	6.26	6.30	6.29	6.74

Yelp, 10 topics
p=0.01 significance at 0.2

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	3.26	3.24	3.27	3.26	3.29	3.27	3.31	3.31	3.24	3.24
T5	3.13	3.13	3.14	3.15	3.18	3.16	3.22	3.23	3.12	3.24
PH	3.21	3.19	3.21	3.21	3.22	3.22	3.24	3.21	3.23	3.31
LO	3.20	3.18	3.18	3.19	3.22	3.19	3.20	3.24	3.22	3.31
P2	3.13	3.10	3.13	3.12	3.16	3.13	3.19	3.22	3.16	3.27
P1	3.15	3.13	3.13	3.15	3.16	3.16	3.22	3.22	3.18	3.29
WL	3.09	3.08	3.11	3.09	3.15	3.12	3.19	3.21	3.15	3.26
SS	3.08	3.09	3.06	3.11	3.13	3.13	3.18	3.21	3.14	3.27
KR	3.07	3.04	3.09	3.08	3.13	3.10	3.18	3.19	3.13	3.24
NS	3.05	3.07	3.08	3.09	3.15	3.13	3.20	3.21	3.13	3.26

Yelp, 50 topics
p=0.01 significance at 0.1

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	5.51	5.49	5.51	5.51	5.48	5.49	5.53	5.51	5.47	5.40
T5	5.30	5.30	5.31	5.34	5.31	5.32	5.40	5.38	5.25	5.47
PH	5.36	5.33	5.35	5.36	5.32	5.31	5.36	5.28	5.38	5.51
LO	5.35	5.33	5.34	5.35	5.31	5.31	5.29	5.36	5.40	5.53
P2	5.26	5.22	5.24	5.25	5.21	5.21	5.31	5.31	5.32	5.49
P1	5.24	5.21	5.23	5.24	5.19	5.21	5.31	5.32	5.31	5.48
WL	5.24	5.23	5.23	5.21	5.24	5.25	5.35	5.36	5.34	5.51
SS	5.17	5.20	5.15	5.23	5.23	5.24	5.34	5.35	5.31	5.51
KR	5.20	5.16	5.20	5.23	5.21	5.22	5.33	5.33	5.30	5.49
NS	5.11	5.20	5.17	5.24	5.24	5.26	5.35	5.36	5.30	5.51

Yelp, 200 topics
p=0.01 significance at 0.09

	NS	KR	SS	WL	P1	P2	LO	PH	T5	T4
T4	7.51	7.49	7.51	7.51	7.49	7.48	7.54	7.54	7.46	7.38
T5	7.18	7.21	7.20	7.24	7.23	7.22	7.33	7.34	7.12	7.46
PH	7.31	7.27	7.29	7.29	7.26	7.25	7.31	7.23	7.34	7.54
LO	7.27	7.25	7.26	7.26	7.24	7.23	7.18	7.31	7.33	7.54
P2	7.15	7.11	7.13	7.13	7.10	7.08	7.23	7.25	7.22	7.48
P1	7.13	7.11	7.11	7.12	7.07	7.10	7.24	7.26	7.23	7.49
WL	7.10	7.09	7.09	7.05	7.12	7.13	7.26	7.29	7.24	7.51
SS	7.01	7.07	7.00	7.09	7.11	7.13	7.26	7.29	7.20	7.51
KR	7.07	7.03	7.07	7.09	7.11	7.11	7.25	7.27	7.21	7.49
NS	6.94	7.07	7.01	7.10	7.13	7.15	7.27	7.31	7.18	7.51

Figure 4: The variation of information between different treatments of corpora indicates that while light stemming may improve the comparative similarity of topic models, heavier stemmers produce less stable topic assignments. The minimum for statistical significance is computed as the maximum $p = 0.01$ value for any topic model as compared with itself (i.e. the 95% confidence interval on the diagonal).

els with fewer topics over Yelp. This does not imply that five-truncation is a competitive stemmer, but rather illustrates that by this measure strong stemmers perform worse than a naive baseline on a corpus with short words and irregular text.

5.5 Influential Words

To identify word types that positively or negatively affect the quality of the model after stemming, we use our idf-probability and entropy metrics for each word type. The idf-probability metric strongly indicates that while conflation improves probability of words on average, the improvement applied primarily to conflated words. Untreated words that do not share a conflation class under a treatment (e.g. "marquess") often become less probable on average after stemming. Their inferred hyperparameters are larger and thus encourage less sparsity in stemmed topic models; as a result, the probability of rarer words in their own conflation classes decreases as that probability is more distributed across topics. This also increases the entropy of stemmed words from a size-one conflation class.

We can confirm several hypotheses from earlier in the paper using these methods. For entropy differences, those conflation classes with the greatest weighted probability improvement for the truncation stemmers in ArXiv include huge conflation classes of words with the same prefix but wildly different roots. In effect, these have forced sparsity where it should not necessarily have been, degrading coherence. As exemplified in the 50-topic NYT models, the Porter stemmer improves the likelihood of common words, like "street" ($TPscore = 5370$) and "mr" ($TPscore = 13945$), an outcome aligned with the rule-based stemmer's aim to cope well with common words. But for rarer words like "purgative" ($TPscore = -17.5$) and "pranks" ($TPscore = -15.4$), no such improvement is seen. These common words do not have extreme entropy values, which supports our hypothesis that while the likelihood of common words improves with Porter stemming, those words were already in the same topic and did not affect model coherence. While we cannot use the same entropy measurement on the context-sensitive lemmatizer, we see the same effect, where the most-improved words are the most common, and the less-likely words in the stemmed model are rare words and names.

Interesting results also arise from the five-truncation stemmer. Unlike prescriptive rule-based stemmers, the truncation stemmer does not produce more errors when typos arise; in fact, it can accommodate typos at the ends of words in a way that other stemmers cannot. While, once again, we observe that the word probabilities of truncated words are much improved for common words and slightly reduced for rare words, we discover that the best entropy improvements from untreated to stemmed words are elongated words and exclamations such as "eeeee" ($\Delta H_w(k) = -2.56$) and "haaaa" ($\Delta H_w(k) = -3.25$). At the opposite score extreme, several classes of words with many misspellings have increased entropy after stemming, but this is potentially misleading; topic models are very good at distinguishing dialects, and systematic misspellings are likely to create differently-spelled but semantically similar topics in a many-topic model. Over one hundred words conflate to "defin" with five-truncation, including upwards of sixty misspellings of "definitely," which removes distinction between good and bad spellers that might be correlated with other features.

6 Related Work

We are not aware of other work evaluating a variety of stemmers and conflation techniques on topic models. Some prior work exists that evaluates the effect of stemming. Several conflation methods were tested on a variety of document clustering algorithms by Han et al. (2012), finding that they could reduce the number of features effectively but that the correct choice of stemmer varied. More recently, Stankov et al. (2013) developed a stemmer to improve document clustering. Both of these techniques demonstrate an improvement in clustering results and a reduction in features required, with the former also introducing the notion of tradeoff between the precision of lemmatization and the efficiency and strength of stemmers.

Additionally, a variety of work exists in the general field of stemmer evaluation, though much of it centers on the information retrieval community. In particular, the work of Harman (1991) highlights some of the fundamental issues of strong stemming, including the potential positive effect of light stemmers like the S-removal stemmer. The notion of stemmer strength is detailed further by Frakes and Fox (2003), as well as several more precise metrics of evaluation of stemmer strength. Survey papers from Jivani (2011) and Rani et al. (2015) detail the different existing stemming and conflation techniques for machine learning applications, including several statistical stemming algorithms that do not rely on a fixed set of rules. Findings suggest that, while these statistical methods have potential, many are inefficient, complex, and difficult to calibrate well enough to produce good results. Though we look forward to seeing the future development of these stemmers, for this work we chose to focus on simpler and more widely used methods.

7 Conclusion

Despite its abiding popularity, stemming does not improve coherence after controlling for the size of vocabulary, and may actually reduce predictive likelihood and increase sensitivity to random initializations. In most cases, the topic model was already grouping together common words with the same root on its own, and gained little by better modeling rare words. Light treatments seem to fare better than strong stemmers, with Krovetz doing particularly well for well-proofread corpora, but the small differences between words that these target such as pluralization and verb conjugation are often already captured by semantic models like LDA.

In certain cases, a stemmer may encode an assumption that is useful for coping with a corpus with heavy variation, as with the 5-truncation stemmer helping to correct misspellings on Yelp. While this does not improve the quality of the topic model by most measures, it may be suited for a particular task involving abnormally varied word forms to which the model is applied. However, for stemmers encoding standard rules of spelling and grammar, such a benefit is unlikely. Given the overly-strong effects of truncation stemming, we suggest using a stemmer as a method of discovering misspellings to fix instead of as a way of repairing them.

A common motivation for stemming is to display more succinct results by not repeating minor morphological variations (such as “place” and “places” in the case of Yelp). As an alternative we suggest *post-stemming* the list of keywords, as shown in Table 4. Stemming a list of top words *after* modeling allows topic models to exploit the nuances of morphologies, such as “apple” and “apples” with respect to the company and the fruit, while still allowing the eventual viewer to browse through the resulting concepts quickly. Post-stemming is computationally much cheaper than stemming the full corpus, requiring only a slightly longer input list of most probable terms. Because context is unavailable for keywords and strong stemmers reduce readability, we would suggest using the S stemmer or a modification of the Porter stemmer to return to English word forms.

Unstemmed	room hotel stay **rooms** pool nice **stayed** strip night bed check clean bathroom desk casino vegas free front resort shower
Stemmed **after** training	room hotel stai pool nice strip night bed check clean bathroom desk casino vega free front resort shower
Stemmed before training	room hotel stai pool nice bed check strip night vega suit casino clean bathroom view desk resort dai walk area

Table 4: An example topic from an unstemmed Yelp 50-topic model with redundant keywords demonstrates that stemming *after* modeling produces the same apparent high-probability words as stemming before.

Vocabulary curation can have a profound effect on the results of statistical models, yet procedures for vocabulary curation have largely been left to unexamined convention and undocumented folk wisdom. We find that a commonly used method, stemming, provides little measurable benefit and may in fact be harmful. As text mining becomes more influential outside core NLP research, more attention must be paid to these issues.

8 Acknowledgements

We would like to thank Jacob Gardner, Jack Hessel, Andrew Loeb, Brian McInnis, and Elly Schofield for helping to refine the writing in this paper. We also would like to thank the TACL editors Mark Johnson and Hal Daumé III and the reviewers for their thoughtful comments and suggestions. The first author was funded by a Cornell University Fellowship.

References

Narayan L Bhamidipati and Sankar K Pal. 2007. Stemming via distribution-based word segregation for classification and retrieval. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):350–360.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media. Available at: http://www.nltk.org/book/.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Matt Chaput. 2010. Stemming library. Available at: https://bitbucket.org/mchaput/stemming.

Antske Fokkens, Marieke Van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st ACL*, pages 1691–1701.

William B Frakes and Christopher J Fox. 2003. Strength and similarity of affix removal stemming algorithms. In *ACM SIGIR Forum*, volume 37, pages 26–30. ACM.

Kuzman Ganchev and Mark Dredze. 2008. Small statistical models by random feature mixing. In *Proceedings of the ACL08 HLT Workshop on Mobile Language Processing*, pages 19–20.

Justin Grimmer and Gary King. 2011. General purpose computer-assisted clustering and conceptualization. *PNAS*, 108(7):2643–2650.

Pu Han, Si Shen, Dongbo Wang, and Yanyun Liu. 2012. The influence of word normalization in english document clustering. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 2, pages 116–120. IEEE.

Donna Harman. 1991. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15.

Carina Jacobi, Wouter van Atteveldt, and Kasper Welbers. 2016. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, 4(1):89–106.

Anjali Ganesh Jivani. 2011. A comparative study of stemming algorithms. *International Journal of Computer Technology and Applications*, 2(6):1930–1938.

Matthew L Jockers and David Mimno. 2013. Significant themes in 19th-century literature. *Poetics*, 41(6):750–769.

Sowmya Kamath S, Atif Ahmed, and Mani Shankar. 2015. A composite classification model for web services based on semantic & syntactic information integration. In *Advance Computing Conference (IACC), 2015 IEEE International*, pages 1169–1173. IEEE.

Robert Krovetz. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202. ACM.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the Association for Computational Linguistics*, pages 530–539.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 366–376. Association for Computational Linguistics.

Siaw Ling Lo, David Cornforth, and Raymond Chiong. 2015. Effects of training datasets on both the extreme learning machine and support vector machine for target audience identification on twitter. In *Proceedings of ELM-2014 Volume 1*, pages 417–434. Springer.

Julie B Lovins. 1968. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. Yass: Yet another suffix stripper. *ACM Transactions on Information Systems (TOIS)*, 25(4):18.

Andrew K McCallum. 2002. Mallet: a machine learning for language toolkit. Available at: http://mallet.cs.umass.edu.

Marina Meilă. 2003. Comparing clusterings by the variation of information. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer Berlin Heidelberg.

Massimo Melucci and Nicola Orio. 2003. A novel method for stemmer generation based on hidden markov models. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*, pages 131–138. ACM.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

Yuhong Nan, Min Yang, Zhemin Yang, Shunfan Zhou, Guofei Gu, and XiaoFeng Wang. 2015. Uipicker: User-input privacy identification in mobile applications. In *24th USENIX Security Symposium*, pages 993–1008.

Chris D Paice. 1990. Another stemmer. *ACM SIGIR Forum*, 24(3):56–61.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Martin F Porter. 2001. Snowball: A language for stemming algorithms. Available at: http://www.snowball.tartarus.org/texts/introduction.html.

SP Ruba Rani, B Ramesh, M Anusha, and JGR Sathiaseelan. 2015. Evaluation of stemming techniques for text classification. *International Journal of Computer Science and Mobile Computing*, 4(3):165–171.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium*, DVD: LDC2009T19.

Ivan Stankov, Diman Todorov, and Rossitza Setchi. 2013. Enhanced cross-domain document clustering with a semantically enhanced text stemmer (sets). *International Journal of Knowledge-based and Intelligent Engineering Systems*, 17(2):113–126.

Chuan Su. 2015. Machine learning for reducing the effort of conducting systematic reviews in SE. *Bachelor Thesis*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Hanna M Wallach, David M Mimno, and Andrew K McCallum. 2009. Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems*, pages 1973–1981.