University of Helsinki

Department of Modern Languages

Language Technology

Bachelor's Thesis

Stemming as a preprocessing step for automatic document timestamp inference

Leo Leppänen

Supervisors:

Graham Wilcock

Atro Voutilainen

March 18, 2015

Contents

1 Introduction

As the amount of free and publicly available information increases, finding relevant information becomes ever more difficult. Through the most common search engines alone, billions of articles are available on the internet (Gulli & Signorini, 2005).

Prior research by Li & Croft (2003) has shown that the relevancy of a search result is often related to a specific point in time: most of the relevant results are clustered around either a past date or the present. It is possible to improve the precision of a query without a large decrease in recall by either limiting the query results to documents from a certain time period or by weighting the result documents based on their closeness to a specified point in time.

The primary problem with this kind of temporal limiting is the difficulty of timestamping the documents reliably. Easily available document timestamps and other metadata are often unreliable or do not in reality specify the document's date of writing. The timestamps of file creation or document indexing both only give an upper limit to the actual creation time of the contents. For example, both will change to a later date when a web page is moved from one location to another. Any timestamps – if present at all – in the document body are often difficult to parse and identify and can also be wrong due to human error.

A method of dating a document without using any outside meta data is therefore clearly needed. The first part of this work examines and compares different content based document dating methods and examines their possible weaknesses. The second part of this work examines an implementation of one of the simpler automatic timestamping methods, and through it the usefulness of stemming as a preprocessing step of dating for agglutinative languages.

2 Methods of timestamp inference

The automatic dating of documents is a relatively new field of study. Due to this only few methods for this type of automatic dating have been presented before. In practice, existing methods can be divided into two primary categories: language model based methods and time expression based methods. Both split the time-line of the corpus into atomic units of the time-line called *chronons* (Alonso et al., 2009). These chronons are the smallest unit of time the algorithm differentiates, for example days. Different time periods are then constructed out of the chronons, for example a certain time period that is a week long would consists of seven chronons, each a day long. The document is then compared to the different time periods and the best match is determined to be the time period the document belongs to.

2.1 Language model based timestamping methods

Among the first studies in the field is de Jong et al. (2005). The method presented by said paper was later named *unigram normalized logarithmic likelihood ratio* (unigram NLLR) by Chambers (2012). Like the name implies, unigram NLLR uses a normilized logarithmic likelihood ratio (Kraaij, 2004) over the document's unigrams (Equation 1). The method determines the likelihood of the document D belonging to the *timespan* Y of the reference corpus C by summing over all the unigrams w in the document D. The document is interpreted as belonging to the timespan Y_d which is the value of Y for which $NLLR(D, Y)$ is maximized (Equation 2).

$$NLLR(D, Y) = \sum_{w \in D} P(w|D) \times \frac{P(w|Y)}{P(w|C)} \tag{1}$$

$$Y_D = \arg\max_Y NLLR(D, Y) \tag{2}$$

Depending on the interpolation method, between 20–24% of the documents can be correctly dated to three month timespans using unigram NLLR. As a totally random dating method would have achieved an accuracy of around 4% using the same date, unigram NLLR is four to five times more accurate than purely random dating.

The unigram NLLR method was further improved by Kanhabua & Nørvåg (2008, 2009). Chambers (2012) calls this improved method *filtered NLLR*. The name filtered NLLR comes from the usage of *term frequency–inverse document frequency (tf-idf)* to filter from the document a subset of N_t most representative words for use in the actual dating (Kanhabua & Nørvåg, 2008).

Tf-idf (Salton et al., 1975) assigns a document's term's representativeness a numeric value by first calculating the terms frequency in the document and then calculating the inverse of the term's frequency in the corpus, called the idf-value.

$$(IDF)_k = \lceil \log_2 n \rceil - \lceil \log_2 d_k \rceil + 1 \tag{3}$$

Where $(IDF)_k$ is the IDF-value for the term k, n is the amount of documents present in the corpus and d_k is the frequency of the term k in document d.

Finally, the term's frequency is weighted by the IDF value

$$TF\text{-}IDF_i^k = f_i^k \cdot IDF_k \tag{4}$$

Where IDF_k is the IDF-value of the term k and f_i^k is the frequency of the term k int the document i.

The equations can also be written more concisely as

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t} \tag{5}$$

Where t is the term being evaluated, d is the document the term belongs to, $tf_{t,d}$ is the frequency of term t in the document d and $df_{t,d}$ is the amount of documents containing the word t within the corpus.

Kanhabua & Nørvåg (2008) describe taking advantage of a multitude of semantic preprocessing methods like collocation extraction, word sense disambiguation, concept extraction, Part-of-Speech tagging and word filtering. The way these methods were used, however, is not clear from their article and the effect of these methods is hard to estimate separately from the effects of their other improvements like usage of Temporal Entropy and more advanced interpolation methods.

Another study by Dalli & Wilks (2006) used statistical language models. The study examined the probability distribution of the document's words belonging to different timespan and used those probabilities to determine the most likely time of writing. This approach is highly similar to the two presented above.

Also of note are the previous works by Kumar et al. (2011, 2012). Both of these are methodologically almost identical to previous works like de Jong et al. (2005) and Kanhabua & Nørvåg (2008, 2009) except for the usage of KL-divergence (Kullback & Leibler, 1951) - of which NLLR is a variant - and a different smoothing mechanism. What sets both studies apart from those mentioned previously, is their larger scale: Kumar et al. (2011, 2012) use Wikipedia biographies, Wiki-year pages and Project Gutenberg short stories, the creation dates of which spread over a much longer time than those of the corpuses used by other studies.

2.2 Time expression based timestamping methods

Aside from language model based timestamping, the other major research direction is time expression based language modelling. This field of study attempts to scour the document for temporal expressions that can be linked to a known point in time and therefore used to date the document to a certain time period.

Chambers (2012) describes a method of searching the document body for time expressions such as "*since* 1991" or "*not before* 2005", which are then used to deduct upper and lower limits for the creation of the document. Together with using a MaxEnt model this time expression analysis, Chambers reports a 77% relative increase in accuracy over the Filtered NLLR method of (Kanhabua & Nørvåg, 2008, 2009).

2.3 Major weaknesses of language model based dating methods

The aforementioned language model based dating methods are a special case of the general document categorization problem and therefore require a similarity in content between the reference corpus and the document being dated. De Jong et al. (2005), Kanhabua & Nørvåg (2008) and Dalli & Wilks (2006) all examined newspaper articles. As Chambers (2012) points out, this introduces a high likelihood of overlap in the material due to multiple newspapers publishing similar articles on the same or close dates. Newspaper articles also use very established and homogeneous language. This makes the results unlikely to apply to the broader scale of writing styles present in the internet and other repositories of knowledge.

A further problem for methods based on language models is the fact that they rely on a reference corpus that is by definition an incomplete and

imperfect representation of all documents written in a certain time period. Due to this imperfection, all results that are based on this reference corpus contain by default some amount of error. Therefore ensuring that the reference corpus being used is of proper quality is of utmost importance.

A third problem arises from the NLLR equations (Equations 3–5). In a situation where the observed unigram does not occur in the reference corpus ($P(w|C) = 0$) the equation is undefined due to division by zero. Kanhabua & Nørvåg (2008, 2009) solve this problem by interpolating a small non-zero probability for all words missing from the reference corpus. This interpolation - and in some cases extrapolation - can distort the results. This distortion is further intensified by the imperfection of the reference corpus. While no studies on the quantity of this distortion were found while writing this thesis, the distortion cannot be wholly ignored.

A final problem of using stemming as a preprocessing step is mainly relevant for agglutinative languages and is discussed in detail in the next chapter.

3 Stemming as a preprocessing step

All language model based methods presented above utilize stemming as a preprocessing step. This preprocessing makes the resulting language model simpler and therefore allows for better results in case of rather limited reference corpuses. The logic behind this approach is that the significance of both unigrams *tsunami* and *tsunamis* is the same if we are merely attempting to date the document. Therefore it would not be beneficial for us to distinguish between these two tokens in our language model.

However, when considering agglutinative languages such as Finnish, it is possible to come up with examples of situations where stemming removes relevant information (Figure 1).

1.1 *Presidentti Niinistö lähti tänään Mäntyniemestä Sotšiin*

1.2 *Presidentti Niinistö lähti tänään Sotšista Mäntyniemeen*

1.3 **Presidentti Niinistö lähteä tänään Mäntyniemi Sotši*

1.4 **Presidentti Niinistö lähteä tänään Sotši Mäntyniemi*

Figure 1: Examples of cases where stemming removes relevant information.

The difference between the unstemmed sentences 1.1 and 1.2 is clear and relevant from a dating point of view. Between the stemmed sentences 1.3 and 1.4 no such difference can be observed, since both sentences could have had the meaning of either 1.1 or 1.2. Therefore in this case stemming actually reduces the amount of relevant information available for the dating process.

For the stemming to be helpful, this information reduction needs to have a smaller negative effect on the accuracy than the accuracy increasing effect of the stemming. While this claim seems plausible enough in context of less-agglutinative languages like English, its plausibility concerning Finnish and other agglutinative languages seems dubious.

Also of note here is that the information lost upon stemming in figure 1 is of the kind that is lost in any case in English texts. This loss of information is due to the bag-of-words approach taken by the algorithm towards the documents: all the sentences 2.1–2.3 will result in the same bag-of-words (sentence 2.4). That is, if we consider the sentences to be sets consisting of all the unique words in the sentences, then the sets for all the sentences 1 to 3 in figure 2 are equal.

2.1 *President Niinistö left today from Mäntyniemi for Sotši*

2.2 *President Niinistö left today from Sotši for Mäntyniemi*

2.3 *President Niinistö left today for Sotši from Mäntyniemi*

2.4 {*President*, *Niinistö*, *Sotši*, *Mäntyniemi*, *left*, *today*, *for*, *from*}

Figure 2: Examples of cases where stemming does not remove relevant information.

It is also important to recognise the fact that a meaning can be expressed in multiple ways by sentences consisting of same root forms but different inflected forms. An example of this would be the sentences *Hän tuli kaupasta* and *Hän tuli kaupoilta*, which both have the same stems, but would result in different bags of words without stemming.

This presents an alternate force to the loss of information caused by stemming. It is not immediately obvious if the increase in accuracy from these cases is enough to compensate for the loss of information.

4 Research question

Based on the above observations on the effects of stemming and the bag of words approach taken by the algorithm, it is hypothesised that the effect of stemming as preprocessing step for English and other less-agglutinative languages is negligible. It is also hypothesized that the effects of stemming are greater for more-agglutinative languages such as Finnish.

This paper attempts to determine the answers to the following questions:

1. Does stemming as a preprocessing step have an effect on the accuracy of dating English language documents?
2. Does stemming as a preprocessing step have an effect on the accuracy of dating Finnish language documents?
3. What is the *direction* of the effect of stemming as preprocessing step for the dating of documents written in agglutinative languages?

5 Methodology

5.1 The programs

The author had previously implemented the de Jong et al. (2005) method as a small, interactive, Java program (Leppänen, 2014a). The method was improved by using a *tf-idf* function to determine the most representative tokens for the document to date, as demonstrated in Kanhabua & Nørvåg (2008).

The program knows two different preprocessors. The "Simple" preprocessor is language independent and merely changes the documents to uppercase, removes all punctuation and changes all numerals to the string "NUMERAL".

The second preprocessor "Snowball" utilizes the Java version (Boulton, 2002) of the Snowball stemmer by Porter (2001). The preprocessor first completes the same steps as the Simple preprocessor and then uses the Snowball to stem all the words in the document.

This study uses a slightly modified version (Leppänen, 2014b) of the original program, that contains functionality for automatically running multiple cross validated runs over a single corpus that contains known dates for each document. Using the default values of a single ten-times cross validated run, the program first shuffles (in memory) the input corpus and then splits it evenly to ten parts. Next, each of the ten parts is used as a test corpus against a training corpus consisting of the nine other parts. In case of multiple runs, this simple process is repeated. After each run, a statistic of the correctly and incorrectly dated documents is printed out.

The program does not utilize the document dates in the corpus as a part of the algorithm, but merely to collect statistics on whether a given document was dated correctly by the algorithm or not.

5.2 The corpuses

To verify the effects of stemming on English, a processed subset of the Reuters-21578 corpus (Lewis, 1997) was used as both reference corpus and test data for English. The modified corpus consisted of $n_d = 15\ 000$ documents. This corpus was used as input for ten ten-times cross validated runs of the program.

For Finnish, the Helsingin Sanomat 1997 corpus (Institute for the Languages of Finland and CSC - Scientific Computing Ltd.) was used. This corpus has $n_d = 39\ 887$ documents and consists of $n_t = 8\ 224\ 152$ tokens. This corpus was used as input for four ten-times cross validated runs of the program.

6 Results

6.1 Stemming and English

The results between the Simple preprocessor and the Snowball preprocessor are extremely similar (Figure 3, Table 1). The results go as far as to show a statistically significant *decrease* in accuracy when the documents are stemmed in the case of daily timespans (Tables 2 and 3). This decreasing effect seems to lessen as timespan length increases, and completely disappears with biweekly and monthly timespan lengths.

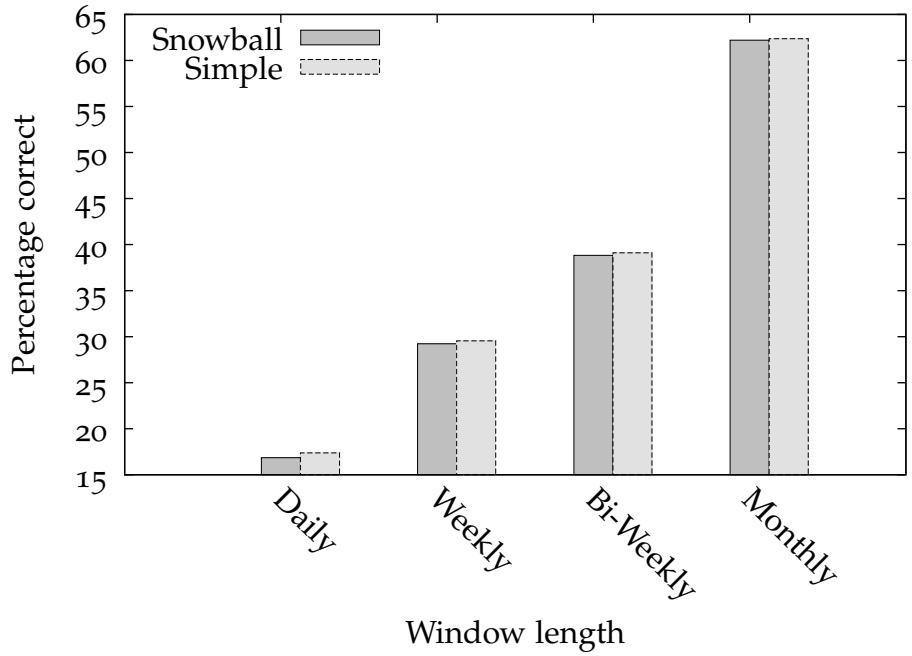

Figure 3: Accuracies of the Simple and Snowball dating methods using different timespan lengths on the English corpus.

Timespan length	**Daily**		**Weekly**		**Bi-weekly**		**Monthly**	
Method	Snowball	Simple	Snowball	Simple	Snowball	Simple	Snowball	Simple
c/n	0.1686	0.1738	0.2923	0.2954	0.3883	0.3911	0.6220	0.6236
σ_M	0.009	0.009	0.011	0.010	0.012	0.011	0.014	0.013

$$c/n = \frac{|\text{correctly dated}|}{|\text{ total}|}$$

σ_M = Standard error

Table 1: Accuracies of the Simple and Snowball dating methods using different timespan lengths on the English corpus.

It seems that stemming as a preprocessing step for dating English language documents is either irrelevant or even detrimental from an accuracy point of view.

Timespan length	Daily	Weekly	Bi-weekly	Monthly
random	0.0238	0.1000	0.1667	0.2500
simple	0.1738	0.2954	0.3911	0.6236
snowball	0.1686	0.2923	0.3883	0.6220

Table 2: Accuracy of NLLR vs. pure chance as evaluated on the English corpus.

Timespan length	Daily	Weekly	Bi-weekly	Monthly
t	4.124	2.1157	1.737	0.8362
df	197.961	197.633	194.874	197.674
p	0.00005	0.03562	0.08397	0.404

t = t-value, df = degrees of freedom, p = p-value

Table 3: Welch Two Sample t-test results between Simple and Snowball preprocessors for each timespan on the English corpus.

These results therefore show that the hypothesis of stemming English documents having a negligible or in some cases even detrimental effect is correct.

6.2 Stemming and Finnish

The results (Figure 4, Table 4) for the Finnish language corpus also correspond to the hypothesis presented in section 4. As stemming causes more changes in the Finnish language documents than in English language documents, this finding is relatively unsurprising.

A rather more interesting result of this experiment is the fact that stemming as a preprocessing step actually improves the accuracy of the dating in a statistically significant fashion (Tables 5 and 6).

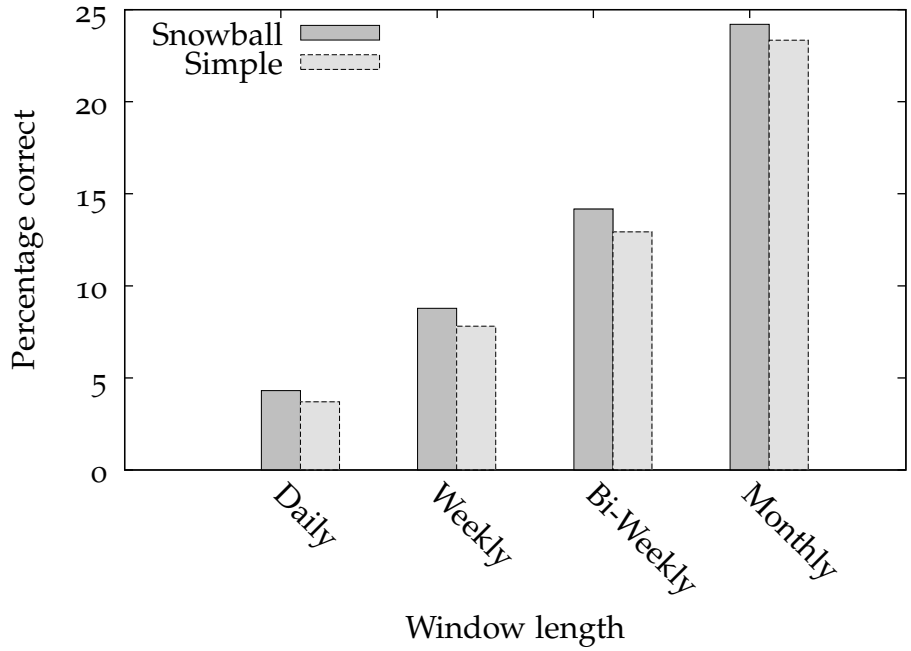

Figure 4: Accuracies of the Simple and Snowball dating methods using different timespan lengths on the Finnish corpus.

Timespan length	**Daily**		**Weekly**		**Bi-weekly**		**Monthly**	
Method	Snowball	Simple	Snowball	Simple	Snowball	Simple	Snowball	Simple
c/n	0.0431	0.0370	0.0878	0.0781	0.1417	0.1293	0.2420	0.2334
σ_M	< 0.001		< 0.001		< 0.001		< 0.001	

$$c/n = \frac{|\text{correctly dated}|}{|\text{total}|}$$

σ_M = Standard error

Table 4: Accuracies of the Simple and Snowball dating methods using different timespan lengths on the Finnish corpus.

Timespan length	Daily	Weekly	Bi-weekly	Monthly
random	0.0029	0.0192	0.0384	0.0833
simple	0.0370	0.0781	0.1293	0.2334
snowball	0.0431	0.0878	0.1417	0.2420

Table 5: Accuracy of NLLR vs. pure chance as evaluated on the Finnish corpus.

Timespan length	Daily	Weekly	Bi-weekly	Monthly
t	8.1382	10.5516	9.8288	6.2464
df	77.963	76.93	76.553	75.964
p	5.094e-12	< 2.2e-16	3.256e-15	2.224e-08

t = t-value, df = degrees of freedom, p = p-value

Table 6: Welch Two Sample t-test results between Simple and Snowball preprocessors for each timespan on the Finnish corpus.

In any case, the results show that the hypothesis of stemming having a substantial effect on Finnish documents is correct. Further more, the results clearly show that the effect is such that stemming of Finnish documents is beneficial for the accuracy of the dating.

7 Conclusions and discussion

The research presented in this paper seeks to answer whether stemming as a preprocessing step has an effect on the accuracy of determining timestamps for English language and Finnish language text. Furthermore, the paper seeks to determine the direction of stemming on the accuracy of determining timestamps for Finnish language texts.

The results presented in the previous chapter show that stemming as a preprocessing step for the dating of English documents is either of no statistical significance or alternatively it has a statistically significant detrimental effect on accuracy. Likewise the results shows that stemming as preprocessing step for the dating on Finnish documents is of statistically significant effect. This effect was determined to be an increase in accuracy for all timespan lengths when stemming was utilized.

Of note is that the nature of both corpuses used in this paper is such that they represent only a part of a single year instead of spanning multiple years. This means that they do not inflict the accuracy lessening effect of cyclic events on the dating in the same way a multi-year corpus would. While this causes our accuracy measurements to differ from those that would have been acquired using larger multi-year corpuses, this absolute difference doesn't affect the *relative* accuracies between our preprocessing methods and therefore we are reliably able to answer our research questions.

In general, the results also confirm that the NLLR-based method used here is - as expected - better than pure chance. Still, especially as we consider smaller timespan lengths, the method used on this paper should probably be considered infeasible for real world usage when dating to timespans with length smaller than a month. This due to the low accuracy the used methods provide when used with short timespan lengths.

Previous research, as discussed in section 2, has already shown possible improvements. These improvements, however, would also be dampened by the cyclic nature of many events such as Christmas, midsummer, Olympics, elections and so on. It's therefore not immediately obvious if the "classic" language model based methods can be improved to provide sufficient accuracy for real world applications where timespan lengths are small. The author speculates that a timespan length of three months – a quarter of a year – would provide sufficient accuracy real world usage.

Future work

This study used the Snowball stemmer (Porter, 2001; Boulton, 2002) which has a somewhat lackluster performance when stemming Finnish. A further study should evaluate the effects of replacing Snowball with *Helsinki Finite-State Transducer* (Lindén et al., 2009) or some other more sophisticated system.

While the results here are clear as far as English and Finnish are considered, another further avenue of research would be to conduct a larger study with more languages comprising a larger spread over the analytic – agglutinative language spectrum, with an aim of finding if the results presented in this paper generalise. If they do, it would be interesting to determine the point (or area) on this spectrum where the change from beneficiality to non-beneficiality occurs.

List of Figures

List of Tables

References

Alonso, Omar, Michael Gertz & Ricardo Baeza-Yates 2009. Clustering and exploring search results using timeline constructions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 97–106. ACM.

Boulton, Richard 2002. Snowball for Java. `http://snowball.tartarus.org/`.

Chambers, Nathanael 2012. Labeling documents with timestamps: Learning from their time expressions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 98–106. Association for Computational Linguistics.

Dalli, Angelo & Yorick Wilks 2006. Automatic dating of documents and temporal text classification. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 17–22. Association for Computational Linguistics.

Gulli, A. & A. Signorini 2005. The indexable web is more than 11.5 billion pages. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, pages 902–903. New York, NY, USA: ACM. URL `http://doi.acm.org/10.1145/1062745.1062789`.

Institute for the Languages of Finland and CSC - Scientific Computing Ltd.Helsingin sanomat 1997. `http://www.csc.fi/`. An electronic document collection of the Finnish language containing 8 million words.

de Jong, FMG, Henning Rode & Djoerd Hiemstra 2005. Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences.

Kanhabua, Nattiya & Kjetil Nørvåg 2008. Improving temporal language models for determining time of non-timestamped documents. In *Research and Advanced Technology for Digital Libraries*, pages 358–370. Springer.

Kanhabua, Nattiya & Kjetil Nørvåg 2009. Using temporal language models for document dating. In *Machine Learning and Knowledge Discovery in Databases*, pages 738–741. Springer.

Kraaij, Wessel 2004. Variations on language modeling for information retrieval.

Kullback, Solomon & Richard A Leibler 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86.

Kumar, Abhimanu, Jason Baldridge, Matthew Lease & Joydeep Ghosh 2012. Dating texts without explicit temporal cues. *arXiv preprint arXiv:1211.2290*.

Kumar, Abhimanu, Matthew Lease & Jason Baldridge 2011. Supervised language modeling for temporal resolution of texts. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2069–2072. ACM.

Leppänen, Leo 2014a. NLLR – A Java program for automatic dating of documents. `https://github.com/ljleppan/nllr`.

Leppänen, Leo 2014b. Nllr4j. `https://github.com/ljleppan/nllr4j`.

Lewis, David D 1997. Reuters-21578 text categorization test collection, distribution 1.0. *http://www. research. att. com/~ lewis/reuters21578. html*.

Li, Xiaoyan & W. Bruce Croft 2003. Time-based language models. In *Proceedings of the Twelfth International Conference on Information and*

Knowledge Management, CIKM '03, pages 469–475. New York, NY, USA: ACM. URL http://doi.acm.org/10.1145/956863.956951.

Lindén, Krister, Miikka Silfverberg & Tommi Pirinen 2009. HFST tools for morphology – An efficient open-source package for construction of morphological analyzers. In *State of the Art in Computational Morphology*, pages 28–47. Springer.

Porter, Martin F 2001. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/.

Salton, G., A. Wong & C. S. Yang 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620. URL http://doi.acm.org/10.1145/361219.361220.