Computer-aided dictionary making: An efficient dictionary construction technique for content analysis

Kohei Watanabe
London School of Economics
and Political Science
k.watanabe1@lse.ac.uk

Abstract

Dictionary-based content analysis has long been popular in social sciences, but manual construction of dictionaries is costly. Latent Semantic Scaling (LSS) is a computer-aided technique for dictionary construction, with which users can produce valid content analysis dictionaries with either 10 to 20 exemplary 'seed words' or about 10 manually scored documents. In this paper, political science examples show that the accuracy of computerized content analysis with LSS dictionaries is comparable to manually compiled dictionaries. R implementation of LSS is also publicly available.

1 Introduction

Use of keyword dictionaries, such as the General Inquirer Dictionary (Stone et al., 1966), LIWC (Francis and Pennebaker, 1993), the Regressive Imagery Dictionary (Martindale, 1975) and DICTION (North et al., 1984) has long been a popular approach to computerized content analysis. The technological simplicity of dictionary-based content makes its use intuitive for non-expert users and it is portable across different platforms.

In the dictionary-based approach, accuracy in computerized content analysis is achieved by careful choice of entry words. A good political science example is the policy position dictionary compiled by Laver and Garry (2000). Despite the fact that the dictionary was created in the 1990s with words chosen by the authors from British party manifestos, it was able to accurately locate the economic policy positions of the Conservatives (Con), the Liberal Democrats (LD) and Labour (Lab) in the 2000s (Figure 1). The correlation between machine and expert scores was as high as r=0.843.

However, valid content analysis dictionaries are only available for a very limited range of topics or types of documents. If existing content analysis dictionaries are utilized for an analysis of documents distinct from these, it raises concerns regarding the validity of results (Grimmer and Stewart, 2013). For example, the Lexicoder Sentiment Dictionary (LSD) successfully measured positive-negative tones in newspaper coverage to predict the outcome of the 2006 Canadian federal election (Young and Soroka, 2012), but it was not able to analyze British political parties' sentiments toward immigration policy as expressed in their 2010 manifestos (Figure 2). The correlation between crowd-sourced coders (Amazon MT) and LSD is only r=0.102.

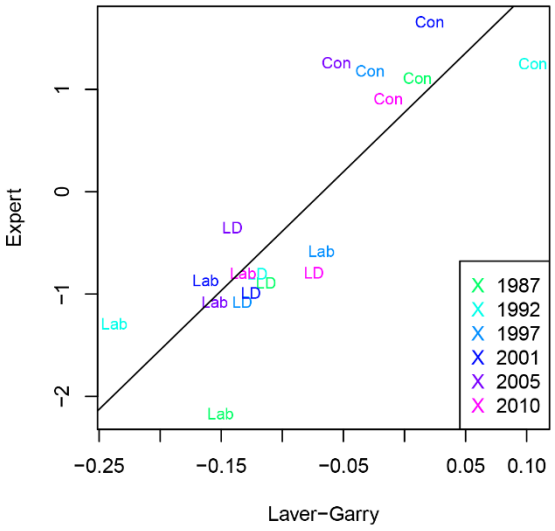

Figure 1: Economic policy position in 1987-2010 UK manifestos by Laver-Garry dictionary.

The inability of LSD to analyze sentiment toward immigration policy is due to the difference in vocabulary between newspaper articles on general elections in Canada and political pamphlets on

immigration policy in Britain. When existing dictionaries appear unsuitable for an analysis of documents of interest, a new dictionary has to be created, but it usually requires a much time and labor, undermining the very benefit of computerized content analysis.

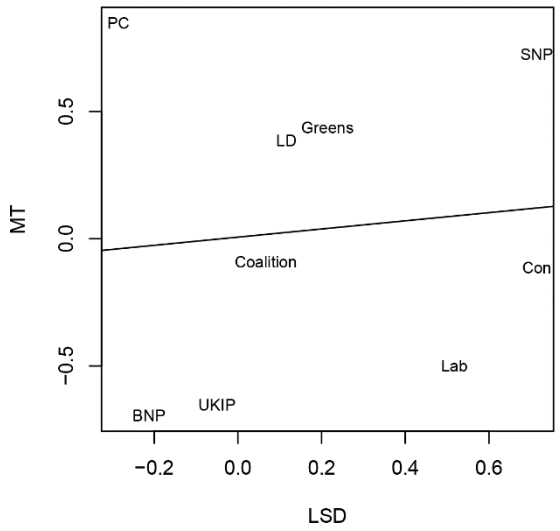

Figure 2: Immigration sentiment in 2010 UK manifestos by LSD

2 Computer-aided dictionary construction

LSS assists construction of subject-specific content analysis dictionaries based on statistical analyses of large corpora. It can be used either as (1) a lexicon-expansion technique or (2) a supervised document scaling technique. Its dictionary construction is based on the following four steps.[1]

2.1 Corpus preprocessing

LSS utilizes subject-specific large corpora to statistically estimate semantic values of words. The minimum size of a corpus for LSS is around 10 million words. In the corpus, documents have to be unitized into sentences, and all the proper nouns and function words should be removed before processing.

2.2 Word selection

LSS selects words that frequently occur with target words, aiming to collect modifiers of the target words, such as 'economy' or 'immigration'. Word selection is performed by collocation analysis of the corpus, and words that appear statistically significantly ($p<0.001$) more frequently than expected enter the dictionary. Collocation is defined as occurrence within 10-word windows from target words and measured by likelihood ration statistic (Hoey, 2012).

2.3 Word scoring

Entry words are scored by cosine similarities to pre-defined 'seed words'. For example, English positive and negative seed words are {good, nice, excellent, positive, fortunate, correct, superior} and {bad, nasty, poor, negative, unfortunate, wrong, inferior} (Turney and Littman, 2003).

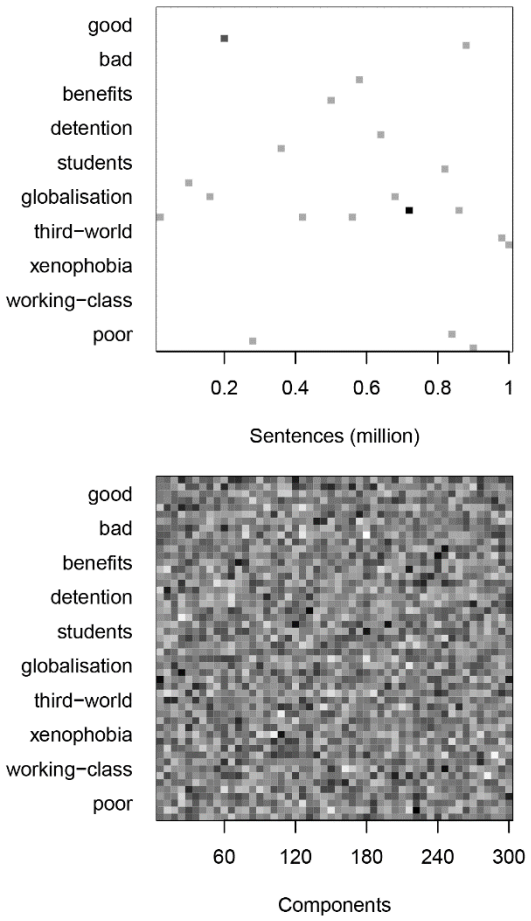

Figure 3: Notional illustration of dimension reduction by Singular Value Decomposition.

Yet, in LSS, cosine similarities are not calculated in the raw term-sentence matrix, but in a reduced term-sentence matrix utilizing Singular Value Decomposition (SVD) to decompose a large sparse matrix into a smaller dense matrix (Figure 3), a technique known as Latent Semantic Analysis (Deerwester et al., 1990). When X denotes the term-sentence matrix, SVD decomposes it into three matrices, U, D and V.

[1] Available in R at https://github.com/koheiw/LSS.

$$X \approx \hat{X} = UDV' \tag{1.1}$$

$$\hat{S} = UD \tag{1.2}$$

With the matrix $\hat{S}$, LSS estimates the sentiment of words by their cosine similarity to the seed words: The sentiment score v_i for a word w_i is a total cosine similarity to seed words weighted by seed scores p_j, which were simply +1 for the positive seed words and −1 for the negative seed words. Here $\cos(w_i, s_j)$ denotes cosine similarity between two row vectors corresponding to entry word w_i and seed word s_j in the matrix $\hat{S}$.

$$v_i = \sum_j^n \cos(w_i, s_j) \cdot p_j \tag{1.3}$$

2.4 Document scoring

Once scores are assigned to entry words, dictionary construction is completed and dictionaries are ready for content analyzing documents. In content analysis, users can either apply LSS dictionaries as (1) words with continuous scores, or (2) words in two discrete categories.

With continuous scores, document scores are weighted means of word scores, as in Wordscore (Laver et al., 2003): when entry words $w_{i...I}$ occur in a document a total of m times, and v_i is the word score and f_i is the frequency count of an entry word w_i, its document score d is computed thus:

$$d = \frac{1}{m} \sum_i^I v_i \cdot f_i \tag{1.4}$$

An LSS dictionary can also be transformed into two sets of words by splitting words by the median score, making its structure identical to traditional content analysis dictionaries. In this case, the document score d is the difference between the normalized frequency of the two sets of words:

$$d = \frac{n_{\text{upper}} - n_{\text{lower}}}{l} \tag{1.5}$$

Where n_{upper} and n_{lower} are numbers of words belonging to the upper and lower half of the dictionary, and l is the total number of words in the document.

3 Example: Immigration sentiment dictionary

With the general English positive-negative seed words, I constructed an immigration sentiment dictionary using LSS without any manual intervention. The corpus for dictionary was British newspaper articles between 2009 and 2010, which contains 15,343 stories or 11.6 million words. Target words were defined by glob patterns, "immingra*" and "migra*".

This immigration sentiment dictionary is comprised of 1,000 words. The most positive and negative words are presented in Table 1. While many of the positive words relate to legal and economic aspects of migration (litigants, detention, benefits, scroungers), negative words mainly concern the social classes and origins of migrants (poor, working-class, frontier, eastern). There are words related to animal migration (conservationist) or migraine (epilepsy, headache), but these words do no harm in analyzing political documents.

Rank	Entry Word	Score
1	issues	0.615
2	policies	0.601
3	ensure	0.585
4	benefits	0.444
5	litigants	0.430
6	huge	0.430
7	detention	0.410
8	wobbling	0.401
9	impromptu	0.396
10	documents	0.390
11	handed	0.374
12	conservationist	0.351
13	joint	0.339
14	restrictive	0.333
15	students	0.326
16	reduced	0.323
17	lounge	0.322
18	bring	0.321
19	major	0.319
20	scroungers	0.313
981	warned	-0.451
982	failure	-0.454
983	areas	-0.457
984	stemming	-0.466
985	makeup	-0.476
986	epilepsy	-0.478
987	countries	-0.506
988	exposed	-0.510
989	eastern	-0.510
990	intentioned	-0.516

991	benefited	-0.527
992	poorer	-0.539
993	frontier	-0.549
994	white	-0.559
995	headache	-0.613
996	negative	-0.646
997	tide	-0.683
998	xenophobia	-0.761
999	working-class	-0.778
1000	poor	-1.302

Table 1: Most positive and most negative entry words for an immigration sentiment dictionary.

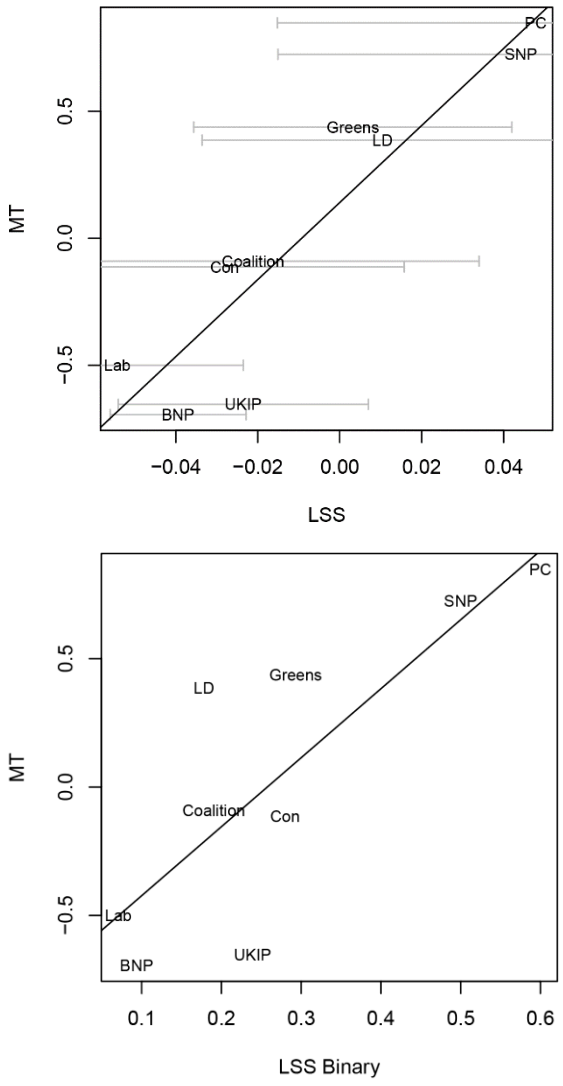

Figure 4: Immigration sentiment in 2010 UK manifestos by LSS.

I applied the immigration sentiment dictionary to the sections on immigration policy in the 2010 UK manifestos. As shown in the first plot in Figure 4, 95% confidence intervals were large due to only brief mentions of immigration in the party manifestos, but point estimation was very accurate (r=0.925). Even when the dictionary was dichotomized by the median score (second plot in Figure 4), it still achieved high correlation with manual scores (r=0.808).

4 Automated seed word selection

Users of LSS can construct their own set of seed words, but selection of seed words for complex dimensions is usually challenging. Therefore, when valid seed words are absent, users can employ a machine learning algorithm to select seed words from a corpus with manually scored documents.

In this automated seed selection, the system tests the suitability of candidates for seed words individually against manually scored documents to obtain polarity of the seed words; then seed candidates are paired with other seed words with opposite polarities to construct a seed set made up of around 10 pairs.

To test the suitability of each seed candidate, the system has to create a large number of tentative dictionaries, but it can be completed very quickly by initial calculation of pair-wise cosine similarities between all of these seed candidates. The system calculates pair-wise cosine similarities in an SVD-reduced matrix $\hat{S}$ (Equation 1.2) that is created from a corpus. The cosine similarities for all pairs are stored in a symmetric matrix D, which has K columns and rows corresponding to the seed candidates $c_{k\ldots K}$. Given the similarity matrix D, a temporary dictionary for a seed word c_k is a kth row or column vector of the matrix D.

$$d_k = D_{\cdot k} = D_{k\cdot} \tag{2.1}$$

First, the system creates K temporary dictionaries in this way, and applies them to the training set (Equation 1.4) to obtain correlation coefficients r_k between scores computed by the temporaries d_k and scores manually assigned. These correlation coefficients allow the system to identify the importance and polarity of the seed candidates. The importance of seed candidates is measured by the sizes of the correlation coefficients; the polarity of seed candidates is given by the signs of the correlation coefficients. The system selects only 50 seed candidates with the largest absolute correlation coefficient from both sides of polarity, and assigns seed scores p_k in the following manner:

$$p_k = \begin{cases} +1, & r_k > 0 \\ -1, & r_k < 0 \end{cases} \quad (2.2)$$

Then, seed words are given adjusted scores to make scoring of documents more consistent when they are combined into a single seed set. An adjusted seed score $\acute{p}_k$is a seed score weighted by the inverse of average squared similarity to other seed candidates in the matrix D (Equation 2.1):

$$\acute{p}_k = p_k \cdot \frac{1}{\sum D_{\cdot k}^2 \cdot \frac{1}{K}} \quad (2.3)$$

Second, with the one hundred seed candidates polarities, the system constructs pairs of seed words $\{c_k, c_l\}$, searching for partner c_l for c_k such that (1) the partner has opposite polarity $p_l \neq p_k$, (2) the dictionary $d_{\{k,l\}}$ yields a higher correlation coefficient than the separate dictionaries $r_{\{k,l\}} > r_k$ and $r_{\{k,l\}} > r_l$, and (3) the correlation becomes the strongest with the partner $r_{\{k,l\}} \geq r_{\{k,\bar{k}\}}$. Starting from the seed candidate with the largest absolute correlation coefficient $|r_k|$, all other seed candidates enter this step-wise paring process. This process continues until at least five pairs have been found; new pairs decrease the overall correlation. The process takes only around 30 seconds on a laptop computer.

In the above process, the system can easily construct a dictionary with a large number of entry words with any set of seed words. Scores assigned to entry words $v_{k\ldots K}$ are calculated simply by taking inner products of the weighted seed scores and a subset of the similarity matrix $\acute{D}$ that only has columns corresponding to the seed words:

$$v_k = \acute{D} \cdot \acute{p}_k \quad (2.4)$$

5 Example: Economic policy position dictionary

As an example of this automated seed word selection, I created an economic policy position dictionary with a corpus of UK economic news stories published prior to elections in 1987, 1992 and 1997, which contains 45 million words in 63,759 news articles. Target words were defined by a glob pattern "economy*". The training set for machine learning was party manifestos from the three pre-millennium elections (9 documents).

In this instance, seed words were selected from words relevant to economy (the same criteria as entry words selection). From the economy-related words, a supervised learning algorithm identified pairs of seed words, producing dictionaries that replicate manual scoring. Through forward step-wise selection, LSS discovered 10 pairs of seed words and assigned weighted seed scores to them (Table 2).

Step	Seed Word	Seed Score
1	rate	478.58
8	run	347.15
10	bottom	191.53
6	miracles	148.99
7	treasury	140.51
9	remain	121.42
3	mpg	110.77
5	tight	107.67
2	acceleration	102.93
4	backdrop	102.84
10	improve	-99.97
8	provide	-109.08
3	generate	-112.00
4	unbalance	-118.95
1	damage	-130.90
7	harm	-131.83
5	based	-137.36
2	appraisal	-148.59
9	disruption	-189.70
6	general	-408.29

Table 2: Seed words for economic policy position dictionary.

The economic policy position dictionary created with the seed words accurately scored not only the British election manifestos from 1987-1997 but also those from 2001-2005, showing its out-of-sample validity (first plot in Figure 5): its errors were smaller than in the Laver-Garry dictionary (Figure 1) particularity in the extreme ranges, although the 2010 manifestos were not very accurately located. Even when the dictionary was dichotomized, the result remained very similar (second plot in Figure 5).

I also applied a Bayesian model, Wordscore, to the same training set to obtain a benchmark for the supervised LSS. The result in Figure 6 clearly shows Wordscore's inability to accurately score the 2000s manifestos by a model created from the 1987-1997 manifestos. This highlights the advantage of corpus-based dictionary construction by supervised LSS. That is, since words in the LSS dictionary were scored based on statistical

analysis of the large corpus instead of the small training documents, it is unaffected by noise in the training set. The clearest indication of the absence of overfitting is the reasonably large confidence intervals for manifestos from 1987-1997 in Figure 5.

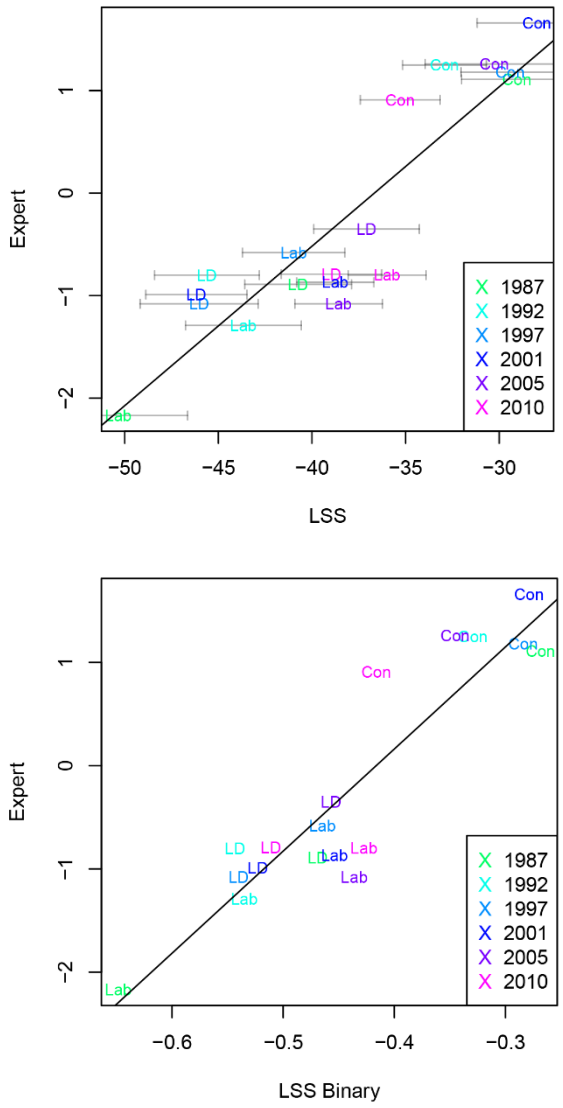

Figure 5: Economic policy position in 1987-2010 UK manifestos by LSS.

Finally, LSS was still not able to score the 2010 manifestos as accurately as the Laver and Garry dictionary, presumably because of the structural break in language of economic policy after the 2008 economic crisis. However, its accuracy can be improved by including economic news articles from 2001, 2005 and 2010 to the corpus. A new dictionary constructed with the extended corpus accurately scored manifestos in the 2000s, better distinguishing the Conservative from the Liberal Democrats and Labour in the 2010 manifestos (Figure 7).

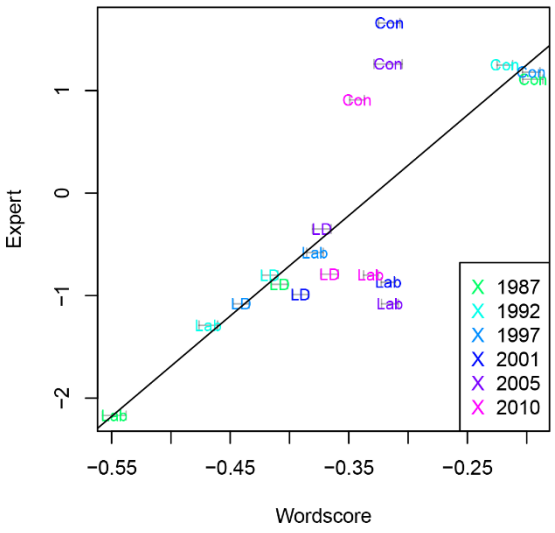

Figure 6: Economic policy position in 1987-2010 UK manifestos by Wordscore.

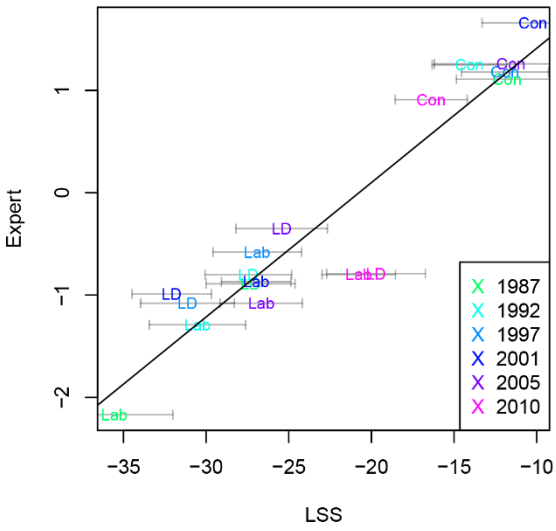

Figure 7: Economic policy position in 1987-2010 UK manifestos by LSS with extended corpus.

6 Conclusion

As evidenced in the examples, LSS dramatically reduces human involvement in dictionary construction: In the lexicon expansion, only 14 manually chosen seed words were required to create a subject-specific sentiment dictionary. In supervised machine learning, only 9 manually scored documents were sufficient for automatically discovering seed words. Further, the accuracy of content analysis using the dictionaries produced

by LSS is comparable to manually compiled dictionaries.

LSS also has an advantage over other supervised techniques that rely on parameter estimation of small training data. By statistically analyzing large corpora, LSS discovers more general semantic values of words, achieving a greater degree of external validity. As a result, LSS dictionaries content analyze unseen documents more accurately than other models.

Reference

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

Martha E. Francis and James W Pennebaker. 1993. LIWC: Linguistic Inquiry and Word Count. Technical report, Southern Methodist University, Dallas, Texas.

Justin Grimmer and Brandon M. Stewart. 2013. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political. *Political Analysis*.

Jesse Hoey. 2012. The Two-Way Likelihood Ratio (G) Test and Comparison to Two-Way Chi Squared Test. *arXiv:1206.4881 [stat]*, June.

Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting Policy Positions from Political Texts Using Words as Data. *American Political Science Review*, 97(2):311–331.

Michael Laver and John Garry. 2000. Estimating Policy Positions from Political Texts. *American Journal of Political Science*, 44(3):619, July.

Colin Martindale. 1975. *Romantic progression : the psychology of literary history*. Hemisphere Publishing ; New York ; London, Washington, DC.

Robert North, Richard Lagerstrom, and William Mitchell. 1984. DICTION Computer Program: Version 1. Technical report, July.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The M. I. T. Press.

Peter D. Turney and Michael L. Littman. 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Trans. Inf. Syst.*, 21(4):315–346, October.

Lori Young and Stuart Soroka. 2012. Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 29(2):205–231.