Language as a Source of Misclassification in the Human Coding of Political Texts*

Tom Paskhalis
London School of Economics
t.g.paskhalis@lse.ac.uk

Christian Mueller
London School of Economics
c.mueller@lse.ac.uk

6th September 2018

Abstract

Designing valid and reliable method for coding large quantities of text is an inherently complicated process. Manual text coding of political manifestos has been shown to be prone to a range of problems that can lead to unreliable results. These include complex coding scheme, intransparent unitization and lack of uncertainty estimates that stem from using only one coder per manifesto. However, the linguistic features that can account for coding errors and the implications of ignoring them have not yet been studied yet. A considerable body of literature in psycholinguistics shows that certain types of sentences are more difficult for humans to process. For example, negative sentences, on average, are harder than their affirmative equivalents. Sentences with main and subordinate clauses appearing in a reversed order might also result in more coding errors. At the same time, current text-as-data approaches work around this limitation of mental syntactic parsing by entirely disregarding word order. In this paper we investigate the effects of negation and main clause position on manual annotation of sentences from party manifestos by crowd-sourced workers. Our results show that both of these features appear to affect coders' ability to discriminate between policy positions. We argue that for measurement models where the theoretical construct can be expected to manifest itself in word frequencies, automated methods outperform humans due to the way language operates.

*We thank Ken Benoit and Ben Lauderdale for helpful comments and discussions on this project. Naturally, errors remain our responsibility.

1 Introduction

Political scientists have long been using content analysis to study texts produced by political actors. Since the original works of Berelson and Lazarsfeld (1948) and Lasswell et al. (1949), researchers turned to systematic reading and coding of texts by human subjects to make inferences about their analytical constructs. Despite the recent advances in computerised text analysis (Grimmer and Stewart, 2013), manual coding remains extremely common technique among applied researchers. The reasons for that are three-fold. First, some analytical constructs accessible to human readers, could still be elusive for machines and, thus, cannot be readily outsourced to computers. E.g. deliberation, conceived as participatory and inclusive debate where rational arguments of all sides can be expressed and considered in a respectful manner, which has generated a large body of both empirical (Fishkin and Luskin, 2005; Steenbergen et al., 2003) and theoretical literature (Dryzek, 2004; Gutmann and Thompson, 2004), has so far been studied exclusively with manual quantitative and qualitative content analysis. Second, although in principle some uses of human coders can be automated, in practice the associated costs might end up being equal or higher than employing research assistants. Named entity recognition and labelling, while being a highly promising technique in natural language processing, requires a large quantity of tailored training data, manually annotated by human coders. The latter can defeat the original purpose of reducing the costs and making research less prone to subjective decisions. And, finally, while using text as their primary source of data and human coding as a chief method of inquiry, certain analytical frameworks can assume that the inferences depend not exclusively on the content, but on a broader political and societal context that has to be taken into account. While some or all these arguments can be germane to a particular subject and topic, quantitative text analysis has been shown to be a valid, reliable and replicable method for estimating party (Laver et al., 2003; Slapin and Proksch, 2008), legislators' (Lauderdale and Herzog, 2016) and interest group (Klüver, 2009) positions, helping predict electoral results (Beauchamp, 2017) and

disentangling the mechanisms of state censorship (King et al., 2013).

The methodological advancement in estimating quantities of interest from textual data has not been met, however, by a theoretical understanding of the processes that allow them to perform so well. Considering the predominant 'bag-of-words' model, that underpins the absolute majority of new methods, most researchers admit that the key assumption made by the model that word order does not matter and can be ignored without losing much information does not reflect how language works. Even more unrealistic assumptions are made by so-called 'topic models' (Blei et al., 2003) that posit that every word comes from a distribution of topics. Thus, several words appearing sequentially in a single sentence can come from multiple topics. A rather non-trivial presumption given our knowledge of linguistics and everyday speech experience. One of potential explanations could be that these modelling simplifications, while counter-intuitive from a linguistic point of view, in practice avoid the pitfalls that human coders might fall into. Although more consistent with our understanding of language, taking text units as a whole inevitably incorporates their ambiguities and complexities. We argue that in cases where the quantity of interest can realistically be assumed to be expressed in word frequencies, automated methods of text analysis can be as accurate or even outperform humans due to the way language operates. Empirically, in this study we test our proposition on Manifesto Project, the largest project in the analysis of political text. Most importantly for our purposes, multiple methods, both involving human coders and automated text analysis have been shown to produce reasonably accurate estimates of party policy positions and their shifts over time.

2 Content Analysis Overview

We start with a brief overview of content analysis. Following Krippendorff (2004), who offers the most comprehensive treatment of the method, content analysis can be divided into 6 steps. (1) *Data collection*, the initial stage of most research, in the case of content

analysis involves gathering the texts that are anticipated to contain the information that can answer researcher's question. With the rise of internet and large-scale projects on digitalisation of paper-based documents, the range of potential sources can vary widely, from newspaper articles to parliamentary speeches to messages on social media. (2) *Unitization* is an important decision of how to divide the collected data into individual units that are then analysed. Typical examples would be entire text, paragraph, sentence and word. The latter being an implied unit in many automated methods of analysis, that can then be aggregated at a higher level of sentence or text. (3) *Sampling* gets far less attention in typical applications of content analysis than it deserves. This puts content analysis in sharp contrast to many other quantitative methods, such as surveys or experiments, where sampling is considered a crucial step in generating valid and reliable estimates of the phenomenon of interest. While conceptually the population of texts to which a researcher makes an inference is a more elusive notion than survey population (Does it encompass all the produced texts that are of interest? Or only recorded? Or even those that could have been generated but were not for various reasons?), we believe it is an important procedure for a carefully designed research that involves content analysis. (4) The *application of analytical framework* roughly corresponds to what Krippendorff calls recording and reducing. In the case of manual human coding this involves getting and training coders, which in a simpler case might mean familiarising them with the codebook and in a more complicated in-depth instruction about the concepts of interest. Subsequently, they are given the texts sampled at the previous step to apply the labels in accordance with the coding instructions. For computerised text analysis at this step the researcher prepares the sampled corpus by putting them in a format required by specific software used for the analysis[1] and then runs the analysis. It is this step that will be the primary focus of our study, as here manual content analysis and automated methods begin to diverge and it is the starting point of much of the discrepancies in the final results. After applying a chosen framework, the method generates some quantities that

[1]This step can include stemming and removing stopwords, the terms that are deemed irrelevant for the analysis.

can then be used to (5) *make inferences* about the studied phenomenon. It is often the case that how inferences are made is, at least, partially determined by the framework. In some cases it could be a word frequency cross-tabulation, in others it would be coefficient estimates from a hierarchical model. (6) And, finally, after interpreting the estimates, researcher *narrates* the results by connecting them back to the literature, context and research question.

The core steps of content analysis (2-4), described above, all treat some segment of text as a unit of analysis. While most automated methods operate on what is called document-term matrix, where rows represent documents and columns record the number of times a given word occurs in that document. Most importantly, the order of rows and columns in the matrix is arbitrary. This drastically contrasts with human coding in that the segments must retain structure to make it possible for humans to read and interpret them. To give a concrete example, the sentence: 'We will not cut benefits.' has to be presented to a human coder intact, while a computer can equivalently read a vector of words: 'benefits', 'cut', 'not', 'we', 'will', where the last three terms are likely to be removed as stopwords. Although, superficially, it might seem that the former contains more information, we argue that this is, ultimately dependent on the type of the quantity of interest. In other words, notions, such as saliency that are likely to be expressed in the repetition of some terms or their combinations, are equally accessible to machines, as opposed to more nuanced concepts such as valence of deliberation. The intuition behind it is that by potentially treating texts more coarsely than human coders, automated methods avoid some of the difficulties inherent to language.

3 Psycholinguistic Foundations

In this study we are focussing on sentence as a unit of analysis. In the context of coding party platforms from their manifestos quasi-sentence has long been a unit choice (Volkens et al., 2013). Quasi-sentences as opposed to natural sentences allow for the possibility

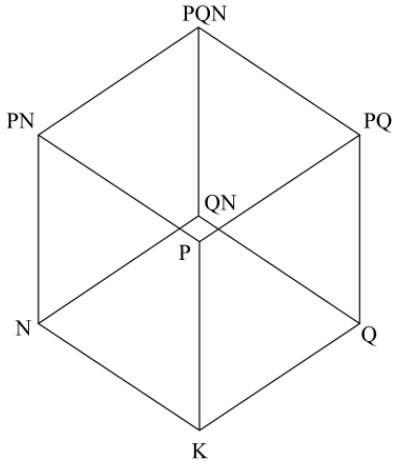
that a single sentence contains several policy propositions that can span multiple codes or even policy domains. Although not unlikely given the language complexity of some party programmes, this has been shown to be prone to arbitrary decisions made by coders while producing no appreciable differences in substantive conclusions (Däubler et al., 2012). Recently, there has been also some progress in political science in grounding this choice not only on the basis of methodological considerations, but also on linguistic theory (Dolezal et al., 2016).

The choice of a unit of analysis is tightly linked to the psycholinguistic origins of our central hypothesis. Since the seminal work of Chomsky (1957), sentence re-emerged as a primary building block of language and unit of perception[2] Chomsky's theory of transformational grammar described sentence as a syntactic configuration generated from kernels by a set of phrase structure rules that constituted the syntax of language. Knowing or learning language in this view, essentially, means possessing or acquiring these rules that allow to distinguish sentences from non-sentences. The idea that not all sentences are created equal followed soon after this theoretical breakthrough. In his landmark article, (Miller, 1962) defined eight types of sentences formed by three possible transformations of the kernel (K) sentence and their combinations: negative (N), passive (P), interrogative (Q), negative passive (NP), negative interrogative (NQ), passive interrogative (PQ) and passive negative interrogative (PNQ).

Figure 1 shows the relationships between all of the eight types by positioning them on the edges of a three-dimensional cube. The crucial insight, confirmed by a number of experimental studies (Gough, 1965; Mehler, 1963) was that the time required to form or process a sentence given another variant of it is a function of their proximity on the cube. Some of the sentence forms, e.g. passive negative interrogative (PNQ) are rather uncommon in written political texts[3]. Other variants, especially the negative, can be expected to occur more frequently. In addition to transformations of simple sentence

[2]Although sentence generated considerable interest among the founding figures of psychology, such as Wilhelm Wundt and William James, it fell into disrepute with the rise of behaviourism. For a comprehensive historical overview see Townsend and Bever (2001).

[3]This, however, might not necessarily apply to speeches.



(a)

K Conservatives will introduce means testing.

N Conservatives will not introduce means testing.

P Means testing will be introduced by Conservatives.

Q Will Conservatives introduce means testing?

NP Means testing will not be introduced by Conservatives.

NQ Won't Conservatives introduce means testing?

PQ Will means testing be introduced by Conservatives?

PNQ Won't means testing be introduced by Conservatives?

(b)

Figure 1: Sentence transformation cube Miller (1962, adapted from figure 12 on p. 760) (a) and a manifesto-type example of its application (b).

structure, composite sentences, formed of several clauses, add another layer of complexity. For example, Holmes (1973) has shown that the order of main and subordinate clauses can equally affect sentence perception. Given the structure of many party programmes and the involvement of humans in parsing and coding sentences, we would expect these cognitive phenomena to affect the resultant estimates. In this study we focus particularly on two syntactic structures most likely to arise in party manifestos: negation and change of sub-clause position.

4 Model of Text Classification

Current approaches to text misclassification tend to focus on the coding scheme and coding instructions that can affect the reliability of estimates (Krippendorff, 2004). In effect, these models posit that by crafting a more balanced and intuitive coding scheme, coupled with clear instructions, researchers can improve the reliability of manually tagging text units.

Mikhaylov et al. (2012) provide the most comprehensive treatment of stochastic misclassification of text units. In this framework misclassification happens through a process

described by the following equation

$$P(T_i^* = j \mid T_i = k) = \theta_{jk},$$

where $j, k = 1, ..., m$ is a code assigned to each text unit out of m categories defined by the coding scheme. θ_{jk} is an off-diagonal cell in the $m \times m$ matrix $\boldsymbol{\Theta}$ that represents text units that were coded as j, given their true label k. Perfect coding would imply that matrix $\boldsymbol{\Theta}$ is a diagonal matrix. Insofar that no real coding is ideal, this matrix would contain off-diagonal elements that result in biased estimates of T_i.

Rewriting the model above as a data-generating process, a text units T_i can be assumed to follow a multinomial distribution

$$T_i \sim \text{Multinomial}(n, \boldsymbol{p}),$$

where $\boldsymbol{p}$ is a vector of parameters over m categories, specified by the codebook. Here the typical goal is to find the population parameters $\boldsymbol{p}$[4] that can be thought as the probabilities for individual text units of coming from a certain category or, equivalently, the proportions of the whole text truly belonging to each category. By manipulating the number and the content of codes, researcher implicitly changes the parameter vector $\boldsymbol{p}$, which can assume different forms. While being mostly theory-driven, certain forms can be more difficult to estimate in practice. More uniform parameter vector, such as $\boldsymbol{p}_3 = (0.3, 0.3, 0.4)$ can be easier to estimate given the same amount of data than $\boldsymbol{p}_3 = (0.05, 0.05, 0.9)$. Although without applying the scheme, it is hard to know what population parameters are, we can assume that an increase in the number of codes will lead to a larger number of categories with very small probabilities, which can result in imprecise estimates. We alleviate this problem by relying on a simplified version of 56-category coding scheme used in Manifesto Project (Volkens et al., 2013), that was adopted by Benoit et al. (2016).

Another implicit assumption of this model of text classification is that each text unit

[4]Assuming $\boldsymbol{\Theta}$ is a diagonal matrix, the vector of interest $\boldsymbol{p}$ can be expressed as $\boldsymbol{p} = (\frac{\theta_{\cdot 1}}{\theta_{\cdot\cdot}}, ..., \frac{\theta_{\cdot m}}{\theta_{\cdot\cdot}})$

has identical parameter vector. However, linguistic and stylistic choices can affect the probabilities of individual text units truly belonging to a certain category. As discussed above, it is not unreasonable to anticipate heterogeneous error due to ambiguity in the sentence. More formally, the vector of probabilities $\boldsymbol{p}$, rather than being a fixed quantity, might itself come from a distribution

$$T_i \sim \text{Multinomial}(n, \boldsymbol{p}_i).$$

Adding another level to the text classification model allows us to reason about the coding process not only in terms of a pre-adopted scheme, but also in terms of linguistic variation that can drive the choice of a specific code. Rather than estimating $\boldsymbol{p}_i$ directly, in the empirical part we are focussing on the effects of negation and sub-clause position on the sentence-level ambiguity that, in turn, can bias the estimates of the quantity of interest if not properly accounted for.

5 Data

The key requirement to be able to assess how language itself might make certain text units more prone to coding errors is that each text unit was coded multiple times by different coders. This requirement is not easy to satisfy with the currently available data as most empirical studies rely on one or, at best, a handful of coders (e.g. Volkens et al., 2016; Dolezal et al., 2016). Although it allows to calculate intercoder reliability on an aggregate level of text, it does not permit the granularity that our hypothesis assumes. In order to test our hypothesis about the effects of linguistic features on misclassification rate, we use the data generated by the crowd-sourced coding of party manifestos in Britain (Benoit et al., 2016). By definition, crowd-source tasks assumes that many readers code multiple text units at least several times. Thus, making this data source come closest to the ideal experimental scenario, where actual sentences could be manipulated.

In order to test for whether there are certain linguistic features which certain mis-

classifications can be attributed to, we use texts from the most coded corpus of political texts, the Manifesto Corpus (Merz et al., 2016) and the efforts in the Manifesto Project (Volkens et al., 2016) more generally. It is the longest, most consistent effort in human coding of political text and provides a good testing ground for the hypothesis. Furthermore, several automated methods have been developed over time for extracting ideological positions and identifying positional shifts between elections (Laver et al., 2003; Slapin and Proksch, 2008; Lo et al., 2016). When considered together, these studies provide substantial evidence that ideological position can be reliably estimated with models based on 'bag-of-words' assumption that word order can safely be disregarded. Unfortunately, the Manifesto Project dataset only contains codings from a single coder (Mikhaylov et al., 2012). This makes it impossible to estimate sentence-level ambiguity parameter and control for coder-specific effects. In order to account for those effects, we rely on a data set of codings of sentences taken from 18 British party manifestos between 1987 and 2010 coded by a small number of expert coders and a large number of crowd workers collected by Benoit et al. (2016).

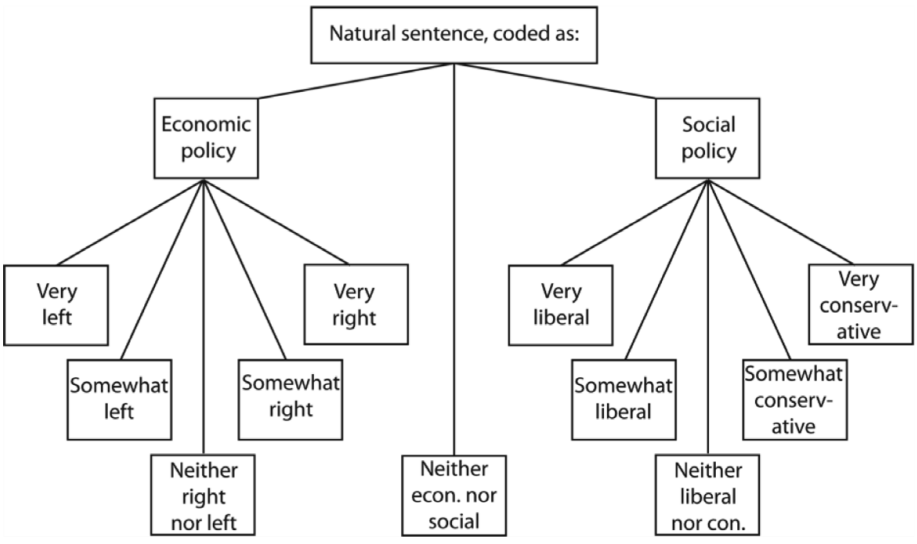


Figure 2: Simplified coding scheme used by Benoit et al. (2016, taken from figure 1 on p. 281).

In these data, coders were individual sentences and they had to code them according to the simplified two-part coding scheme in figure 2. The first part of the task was to

decide whether a sentence contains statements related to economic or social policy. If the policy domain was determined to be either economic or social, the sentence had to be coded on a five-point ordinal scale ranging from 'very left' to 'very right' and 'very liberal' to 'very conservative' for the economic and social dimensions respectively.

Level	Min.	1st Quart.	Median	Mean	3rd Quart.	Max.
Coder	3	10	14	144	69	10180
Sentence	4	5	5	10	15	215

Table 1: Coder- and sentence-level summary statistics of number of codings.

It is plausible, however, that even a limited subset of sentence transformations does not occur often enough in party programmes to make a difference for human coding. Table 2 shows the percentages of sentences in British party manifestos. These percentages are not only non-trivial, ranging from 4% to 10% of all sentences for negation and from 4% to 14% for prepositional phrases, they also covary with changes in electoral politics. We can observe a lagged increase in the number of negative sentences when a party switches from ruling to opposition status and vice versa. Conservatives in their 2001 manifesto after landslide victory by Labour in the previous election, used 3 percentage points more negative sentences in their manifesto than in 1997. This spike tails off, but never goes back to pre-1997 level. Labour, on the contrary, reduce their proportion of sentences containing negation from 10% to 5% between 1997 and 2001. The implications of this association between linguistic features and politically relevant phenomena are two-fold. First, when relying on human readers to code sentences, the texts have to be either tested for their linguistic heterogeneity prior to commencement of coding[5] or these differences have to be modelled explicitly. Second, when not accounted for this differential usage of certain linguistic features can not only bias the estimates, but do so in a way that is hard to predict.

For the main analysis, we focus on the coding decision of the crowd workers and

[5]The emerging literature on linguistic complexity can provide one avenue for such testing (Spirling, 2015).

exclude the codings by expert coders. These data contain 184,328 coding decision for 18,263 sentences made by 1,280 crowd workers. Table 1 contains summary statistics of the number of codings in our the data by coder and sentence. The median, which provides a good estimate for the average number of codings because the distribution of both variables is highly skewed, for coders is 14 codings and for sentences is 5 codings. Thus, each sentence has been coded on average by 5 different coders, each of whom coded on average 14 sentences.

We identify and incorporate three different language features with a potential for effecting misclassification.[6] The first potential source for misclassification is negation as negated statements, e.g. saying 'we do not support' instead of 'we are against', tend to be harder to comprehend than their affirmative equivalents. We automatically extract an indicator for whether the sentence contains negation from the parsed text by looking for dependency relations of type 'negation modifier' (see Choi and Palmer, 2012, 33).

The second and the third potential features for misclassification are related to sentences which contain subordinate clauses. One of them is concerned with prepositional phrases which are placed before the main clause of the sentence. Such phrases violate the standard subject-verb-object ordering of the English language sentence, which might make it harder to comprehend and code the main clause. Such phrases are detected from the dependency parse by looking for prepositions which modify the main verb of the sentence but occur before it. The other potential feature for misclassification is an indicator for subclauses more generally. That is, we include all subclauses independent of where in the sentence they occur and the type of subclause. This feature is extracted by detecting whether a sentence contains a comma which are not used in an enumeration. The following are two example sentences which contain the linguistic features discussed previously:

Prepended prepositional phrase (Liberal Democrats 1997)**:** We will, *in partnership*

[6]We rely on `spacy`, a natural language processing framework in Python, to do the part-of-speech tagging and dependency parsing. See Choi et al. (2015) for an evaluation of the crucial dependency parsing component.

Party	Year	Sentence (N)	Words (avg.)	Negation (%)	Prepositional (%)	Subclause (%)
Conservatives	1987	1015	17.5	6.5	11.4	18.9
Conservatives	1992	1731	17.1	4.9	12.6	21.8
Conservatives	1997	1171	17.9	7.4	10.8	18.4
Conservatives	2001	748	16.9	10.0	8.0	21.4
Conservatives	2005	414	17.0	7.5	12.8	26.6
Conservatives	2010	1240	20.6	8.2	9.3	35.6
Labour	1987	455	20.0	5.1	9.7	19.6
Labour	1992	661	18.7	5.0	12.0	28.4
Labour	1997	1052	16.6	10.3	7.6	19.6
Labour	2001	1752	17.2	5.3	9.2	25.8
Labour	2005	1186	20.1	6.9	14.2	23.9
Labour	2010	1349	21.7	7.2	11.3	35.2
Lib Dems	1987	878	22.2	10.1	10.9	21.5
Lib Dems	1992	884	19.4	7.1	9.6	27.5
Lib Dems	1997	873	16.1	4.4	6.0	19.7
Lib Dems	2001	1178	17.9	6.3	9.3	22.2
Lib Dems	2005	821	19.3	10.2	12.2	31.8
Lib Dems	2010	855	20.8	8.2	7.0	28.9
Total		18263	18.7	7.1	10.3	25.0

Table 2: Descriptive statistics for linguistic structures in UK party manifestos.

with the agriculture industry, draw up a national strategy for farming in order to provide a framework for public policy and private decision-making over the next 10 years.[7]

Subclause with comma (Conservatives 1992)**:** We will reinforce the rights of the individual in the world of work, and break down artificial barriers to advancement.

Table 2 contains an overview of the three potential features effecting misclassification we use in the analysis grouped by manifesto. We also report the total number of sentences per manifesto for reference as well as the number of words per sentence averaged over manifestos. The number of words in a sentence is included as a covariate in the model we propose and estimate below.

6 Modelling Ideological Position and Linguistic Ambiguity

To estimate the misclassification rate, we need to incorporate both substantive quantities of interest that researchers would like to estimate, as well as the effect of linguistic features in a single model. Benoit et al. (2016) propose a modified item response theory (IRT) model, which allows the simultaneous estimation of coder effects, the ideological position of sentences and the tendency of some sentences to be harder to categorise. Their model is a four-dimensional two-parameter Bayesian IRT model with latent dimensions for the decision to (1) choose economic policy domain over none, (2) choose social policy domain over none, (3) choose a more right over a more left code for economic policy, and (4) choosing a more conservative over a more liberal code for social policy (Benoit et al., 2016, 282). The latent outcome μ^* in dimension d for sentence j coded by coder i is then defined by the following IRT model

$$\mu^*_{ijd} = \chi_{id}(\theta_{jd} + \psi_{id}),$$

[7]Prepositional phrase in italic and main verb underlined.

which incorporates a latent position of the sentence θ, a latent position or 'bias' for each coder, and a discrimination parameter χ for each coder which captures coder sensitivity to changes in the latent parameters. These latent outcomes are then translated into coding decision such that μ_1^* and μ_2^* are the propensities to choose the economic and social policy dimensions over none in a multinomial logit model and μ_3^* and μ_4^* are used in an ordinal logit model for choosing higher vs. lower codes in the economic and social policy dimension. As the main goal of this model is to estimate the latent position at the level of the manifesto, it incorporates a hierarchical setup for the sentence-level latent position parameters

$$\theta_{jd} \sim \mathcal{N}(\bar{\theta}_{kd}, \sigma_d^{\theta}),$$

where k indexes the manifesto that contains sentence j.

We propose to extend this model to incorporate misclassification by modifying the discrimination parameter χ_{id}. As briefly mentioned above, the discrimination parameter in the original specification captures the ability of crowd coders to translate changes in the latent position of a sentence into changes in the latent propensity for the multinomial or categorical logit model. We extend this idea by modelling the discrimination parameters as a regression of sentence-level attributes, mainly variables for linguistic features which potentially lead to misclassification, as well as a coder- and sentence-level intercept. Specifically, we propose the following model for the discrimination parameter

$$\chi_{ijd} \sim \mathcal{LN}(\exp(\gamma_{id} + \alpha_{jd} + X_j \boldsymbol{\beta}_d), \sigma_d^{\chi}),$$

where γ_{id} and α_{jd} are the coder- and sentence-level intercepts, X_j is a matrix of sentence-level language features, and $\boldsymbol{\beta}_d$ is a vector of regression coefficients associated with the language variables. As the discrimination parameters are restricted to be positive for location identification of the latent dimensions, we specify a log-normal distribution and additionally exponentiate the mean parameter. Following standard practice in the IRT modelling literature (e.g. Armstrong et al., 2014; Jackman, 2001), we locally identify the

recovered latent space by placing informative priors on the predictors for the discrimination parameter, with

$$\alpha_{jd} \sim \mathcal{N}(0, \sigma_d^{\alpha}) \qquad \gamma_{id} \sim \mathcal{N}(0, 1)$$
$$\boldsymbol{\beta}_d \sim \mathcal{N}(0, 1) \qquad \psi_{id} \sim \mathcal{N}(0, \sigma_d^{\psi}).$$

7 Results

We estimate the model developed above in Stan (Carpenter et al., 2017) using the `rstan` package (Stan Development Team, 2018) in R (R Core Team, 2017). We retain 500 sampling iterations after an initial adaptation phase for the MCMC algorithm of 500 iteration. Figure 3 presents the resulting estimates of the regression coefficients in the misclassification model. Because all of those coefficients were assigned informative standard normal priors for identification purposes, the scale of the resulting coefficients is constrained to this range. Therefore, estimates around -2 and 2 would be extreme negative and positive influences of the associated structure on the discrimination parameter.

Whether a sentence contains a negation only has a significant effect on the discrimination for coding the economic scale. This effects is, however, quite large given the scale of the coefficients discussed above and negative. That is, sentences containing negation have a significantly lower discrimination which implies that the estimated latent coefficient for the sentence is less closely linked to the chosen code on the economic scale.

In a similar vein, prepended prepositional phrases significantly decrease discrimination on the economic scale while there is no significant effect on choosing either the economic or social dimension. Interestingly, prepended prepositional phrases *increase* the discrimination for assigning a position on the social scale. As this goes against the theory of misclassification we discuss above, this results warrants more investigation. A possible explanation might be that there are few cases of sentences coded as containing a so-

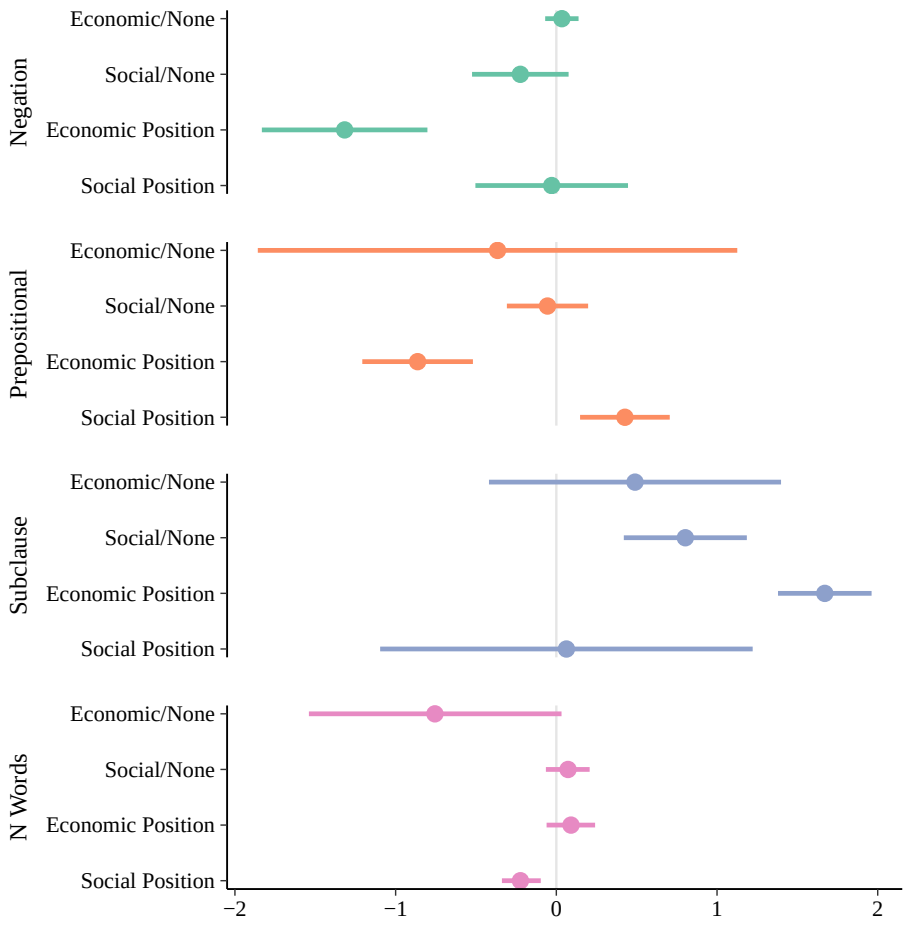


Figure 3: Posterior mean and 95% credible interval for $\boldsymbol{\beta}$ coefficients.

cial policy statement with negation. Also, results from Benoit et al. (2016) suggest that choosing a position on the social scale was generally associated with lower discrimination.

The results for the indicator for containing any subclause are puzzling given our theory. Deciding on whether a sentence contains a social policy statement and choosing a position on the economic scale seems to be significantly easier, on average, for sentences which contain a subclause than sentences without one. One potential explanation is that having a subclause is not a specific enough feature to reliably effect misclassification as about one in four sentences contains a subclause (see table 2).

Another possible explanation would be that it might be easier to assign codes to longer sentences, as they could contain additional information to base the judgement on. Sentences that contain a subclause tend to be longer than sentences that do not. However, the coefficient for the number of words associated with these two dimension is not significantly different from 0. The number of words in a sentence only has a significant effect on coding the position on the social scale. Here, longer sentences seem to be harder to code reliably. It is still possible that raw length of sentence does not capture the additional information added by short qualifying or clarifying subclauses.

8 Discussion

Although preliminary, our results indicate that linguistic features, such as negation and clause position, have some effect on how well humans differentiate between ideological positions. These results are not entirely novel in light of years of research in psycholinguistics and human perception of language stimuli. However, coupled with the fact that those features vary over time and can be associated with electoral victories and entering government, this paints a cautious picture of complete disregarding of language when assigning content coding tasks to humans. We suggest that language homogeneity has to be checked rather than assumed prior to the start of content analysis that relies on human coders. Another possibility that we investigate by using Bayesian hierarchical item

response theory, is to explicitly model these assumptions in a single-step estimation procedure. The results also provide some insight into the success of automated techniques that rely on 'bag-of-words' assumption. We theorise that a reason for their notable performance lies in the benefits that coarsening raw text brings. Although some information is indeed lost when word order is disregarded, in certain applications, especially those that rely on concepts that can be expected to be expressed in word frequencies, this loss is outweighed by removing ambiguity that is inherent to language.

References

Armstrong, II, D. A., R. Bakker, R. Carroll, C. Hare, K. T. Poole, and H. Rosenthal (2014). *Analyzing Spatial Models of Choice and Judgement with R*. CRC Press.

Beauchamp, N. (2017). Predicting and Interpolating State-level Polling using Twitter Textual Data. *American Journal of Political Science 61*(2), 490–503.

Benoit, K., D. Conway, B. E. Lauderdale, M. Laver, and S. Mikhaylov (2016). Crowd-sourced Text Analysis: Reproducible and Agile Production of Political Data. *American Political Science Review 110*(2), 278–295.

Berelson, B. and P. F. Lazarsfeld (1948). *The Analysis of Communication Content*. Chicago: University of Chicago Press.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*, 993–1022.

Carpenter, B., A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell (2017). Stan: A probabilistic programming language. *Journal of Statistical Software 71*(1), 1–32.

Choi, J. D. and M. Palmer (2012). Guidelines for the CLEAR style constituent to dependency conversion. Technical Report 01-12, Institute of Cognitive Science, University of Colorado Boulder, Boulder, CO.

Choi, J. D., J. Tetreault, and A. Stent (2015). It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 387–396. Association for Computational Linguistics.

Chomsky, N. (1957). *Syntactic Stucture*. The Hague: Mouton & Co.

Däubler, T., K. Benoit, S. Mikhaylov, and M. Laver (2012). Natural Sentences as Valid Units for Coded Political Texts. *British Journal of Political Science 42*(4), 937–951.

Dolezal, M., L. Ennser-Jedenastik, W. C. Müller, and A. K. Winkler (2016). Analyzing Manifestos in their Electoral Context A New Approach Applied to Austria, 2002–2008. *Political Science Research and Methods 4*(3), 641–650.

Dryzek, J. (2004). *Deliberative Democracy and Beyond*. Oxford: Oxford University Press.

Fishkin, J. S. and R. C. Luskin (2005). Experimenting with a democratic ideal: Deliberative polling and public opinion. *Acta Politica 40*(3), 284–298.

Gough, P. B. (1965). Grammatical Transformations and Speed of Understanding. *Journal of Verbal Learning and Verbal Behavior 4*(2), 107–111.

Grimmer, J. and B. M. Stewart (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis 21*(3), 267–297.

Gutmann, A. and D. F. Thompson (2004). *Why Deliberative Democracy?* Princeton: Princeton University Press.

Holmes, V. M. (1973). Order of Main and Subordinate Clauses in Sentence Perception. *Journal of Verbal Learning and Verbal Behavior 12*(3), 285–293.

Jackman, S. (2001). Multidimensional analysis of roll call data via bayesian simulation: Identification, estimation, inference, and model checking. *Political Analysis 9*(3), 227–241.

King, G., J. Pan, and M. Roberts (2013). How Censorship in China Allows Government Criticism but Silences Collective Expression. *American Political Science Review 107*(917), 326–343.

Klüver, H. (2009). Measuring Interest Group Influence Using Quantitative Text Analysis. *European Union Politics 10*(4), 535–549.

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology* (2nd ed. ed.). Sage Publications.

Lasswell, H. D., N. Leites, and Associates (1949). *Language of Politics: Studies in Quantitative Semantics*. Cambridge: MIT Press.

Lauderdale, B. E. and A. Herzog (2016). Measuring Political Positions from Legislative Speech. *Political Analysis 24*(3), 374–394.

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

Lo, J., S.-O. Proksch, and J. B. Slapin (2016). Ideological Clarity in Multiparty Competition: A New Measure and Test Using Election Manifestos. *British Journal of Political Science 46*(3), 591–610.

Mehler, J. (1963). Some Effects of Grammatical Transformations on the Recall of English Sentences. *Journal of Verbal Learning and Verbal Behavior 2*(4), 346–351.

Merz, N., S. Regel, and J. Lewandowski (2016, 4). The manifesto corpus: A new resource for research on political parties and quantitative text analysis. *Research & Politics 3*(2), 1–8.

Mikhaylov, S., M. Laver, and K. Benoit (2012). Coder Reliability and Misclassification in the Human Coding of Party Manifestos. *Political Analysis 20*(1), 78–91.

Miller, G. A. (1962). Some psychological studies of grammar. *American Psychologist 17*(11), 748–762.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Slapin, J. B. and S.-O. Proksch (2008). A Scaling Model for Estimating Time-Series Party Positions from Texts. *American Journal of Political Science 52*(3), 705–722.

Spirling, A. (2015). Democratization and Linguistic Complexity: The Effect of Franchise Extension on Parliamentary Discourse, 1832-1915. *The Journal of Politics 78*(1), 235–248.

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.17.3.

Steenbergen, M. R., A. Bächtiger, M. Spörndli, and J. Steiner (2003). Measuring Political Deliberation: A Discourse Quality Index. *Comparative European Politics 1*(1), 21–48.

Townsend, T. G. and D. J. Bever (2001). *Sentence Comprehension*. Cambridge, Massachussets: Massachusetts Intitute of Technology.

Volkens, A., J. Bara, I. Budge, M. D. McDonald, and H.-D. Klingemann (2013). *Mapping Policy Preferences from Texts III: Statistical Solutions for Manifesto Analysts*. Oxford: Oxford University Press.

Volkens, A., P. Lehmann, T. Matthieß, N. Merz, and S. Regel (2016). Manifesto project dataset (version 2016b).