Automatic Estimation of Human Values Using Supervised Machine Learning

髙山, 泰博

https://doi.org/10.15017/1500748

出版情報：九州大学, 2014, 博士（情報科学）, 課程博士
バージョン：

Automatic Estimation of Human Values Using Supervised Machine Learning

February 2015

Yasuhiro Takayama

Contents

Chapter 1

Introduction

1.1 Background and Purpose

Social scientists have used findings from content analysis to provide information for public opinion surveys and decision-making by governmental agencies on contentious issues. Content analysis is typically performed by trained human annotators and is conducted by means analyzing written or transcribed documents. Within these texts, annotators can detect the human values that the writer consciously or subconsciously expressed in textual form (Cheng and Fleischmann 2010; Fleischmann 2014). For example, they analyzed the values of stakeholders who support or oppose the idea of "net neutrality" as expressed in testimonies prepared for public hearings for U.S. legislative bodies and regulatory agencies. This research was based on a subset of the meta-inventory of human values (Cheng and Fleischmann 2010), focusing on the net neutrality debate (Cheng 2012). The human values specified in the inventory include freedom, honor, innovation, justice, social order, and wealth. Content analysis is a type of qualitative analysis, thus typically subject matter experts must conduct the analysis. Hence it is fairly costly and analysis remains restricted because analysis of a large number of documents is not feasible. For example, it takes an hour on average to annotate one document (approximately eighty five sentences with nine hundred words or less) with values, then nearly two months are required to obtain a satisfactory number of annotated documents for content analysis. Therefore, the goal of this research is to explore novel methods for automatically detecting human values invoked in textual documents using cutting-edge innovations in machine learning.

For facilitating content analysis, several inventories of human values are used in social science research (e.g., Friedman et al. 2006; Schwartz 1994). Integrating key components of these studies, Cheng and Fleischmann (2010) defined human values as follows: "values serve as guiding principles of what people consider important in life." And they developed the Meta-Inventory of Human Values (MIHV), which is intended to be applicable to various test collections by selecting values specific to the debate at issue and by iteratively refining annotation guidelines (Fleischmann et al. 2009; Cheng et al. 2012).

Social scientists considered that human values were reflected on statements rather than words, thus traditional paper-based annotations for values could annotated any

length of passages. In addition annotated passages often overlap. Cheng et al. (2012) constrained each annotation to be a single sentence but allow for more than one value per sentence. This set up is well suited to supervised machine learning. They applied MIHV and sentence-based annotation to the net neutrality corpus. After several round of refining the annotation guideline, the resulting 9,890 sentences in the original net neutrality corpus, containing 102 documents, are annotated with zero or more of six human values: *freedom*, *honor*, *innovation*, *justice*, *social order*, *wealth*. We adopt this net neutrality corpus as our test collection because its attractive setup. For making the data more tractable, we remove longer sentences and sentences whose boundaries are uncertain, then stemmed them. Finally, we obtained the remaining 8,660 sentences as data for our experiments.

Ishita et al. (2010) have reported the classification of human values for the earlier round of the net neutrality corpus using *k*-NN (Nearest Neighbor) classifiers for 2,005 sentences over 28 documents. They obtained a macro-averaged F_1 (a harmonic mean of precision and recall) of 0.48 for eight human values. To scale up social science research (Fleischmann et al. 2012), we need to improve the ability to effectively analyze larger data corpora. The lexical features must serve as a basis for classification of human values. We first compared the effectiveness of a wide range of classifiers available within Weka (Hall et al. 2009), and we found that Support Vector Machines (SVMs) (Joachims 1998, 2002) performed best for our corpus.

The human values estimation resembles sentiment analysis, which has been extensively researched (Pang and Lee 2008; Liu 2011). An important difference is that estimation of human values is a multi-label classification, whereas sentiment analysis is typically modeled as a binary classification. Importantly, human values can help to explain sentiment, given their explanatory power in relation to attitudes and behavior (Wiebe 1994). We focus on human values estimation problem with the net neutrality corpus and our modeling of relation between sentence-level labels and word-level human values in this thesis.

We have the following data as some of basic statistics for the net neutrality corpus: the total number of sentences is 8,660 as mentioned before; The number of distinct words is 6,223, the total number of word occurrences is 89,095; and thus the average number of occurrences of words types is 14.3. However, the number of occurrences of two-thirds of distinct words is less than 5. In addition, the average number of words per sentence is only 10.3. (Note that the corpus we use in our experiments actually consist of word stems instead of words as described above, however, we use the term “word” for readability in this chapter.)

That is, we face two kinds of data sparseness for estimating human values in the net neutrality corpus: (a) the number of sentences is small for training classifiers i.e. eight thousand sentences or so; (b) the number of words per sentence is very small, i.e. ten or so within a sentence. We have to accept the above (a) because our purpose is to substitute human annotators with machine classifiers to reduce the costs of annotation. However, it is challenging to estimate the relationships between sentence-level human values and

the words within the sentence for detecting values of the sentence, because most of the words occurred rarely. In addition, the conventional best performed SVMs are general purpose binary classifiers, thus it is unclear whether SVMs with word unigram features can appropriately deal with relationships between the multiple human values annotated in a sentence and words as constituents of the sentence. Therefore, estimation of human values remains a matter of research to explore classifiers with higher accuracy which is specialized for detecting human values.

1.2 Contributions of the Thesis

This section describes the contributions of this thesis for estimating human values.

Firstly, we found that augmenting feature vectors which include hypernyms and synonyms did not substantially contribute to improving classification effectiveness of human values in experiments using SVMs with augmented feature vectors. We tried to use augmented feature vectors to cope with the above data sparseness (b). For detecting sentence-level human values, it is revealed that we could not deal with training data sparsity by augmenting features using word semantic categories, in spite of the fact that word semantic categories are effective for word sense disambiguation or syntactic disambiguation. The fact suggested us the next direction of consideration of a method that assigns the values directly to words, instead of assigning the values to word semantic categories.

Secondly, we formulated the human values for a sentence as an aggregation of human values for words that are constituents of the sentence. Then we proposed a probabilistic latent "value" model, which we call *LVM*, that automatically switches the case where estimation of human values for a word is influenced by the previous word and the case where the estimation is done without influence from the previous word. We achieved that approximately three percent relative improvement in F_1 score by the proposed method, compared with one by the conventional method that used an SVM with simple bag-of-words features, and we confirmed that the improvement is statistically significant. We also demonstrated that the classification effectiveness of *LVM* is comparable to the second human annotator who was well-trained. This means that human annotators might be able to be replaced by classifiers that use our proposed *LVM*.

Thirdly, we built up a "values-dictionary" which consist of words and word pairs with human values with the consideration for the use by social scientists, and we demonstrated the possibility of support them to conduct content analysis. The classification effectiveness of our proposed *LVM* achieved high scores, however, it is difficult for humans to interpret the estimation results because the estimation using *LVM* is done by integrating probabilities of words (or word pairs) representing the values in that sentence. Therefore, we proposed a simplified model of *LVM*, where the model extracts words (or word pairs) with human values as entries of a values-dictionary, that are uniquely determined independently from a specific sentence. The model is an

approximate probabilistic optimization whose objective function is F_1 score for detecting sentence-level human values. In addition, we have provided the values-dictionary to social scientists for content analysis, and we convinced that the values-dictionary is usable for a tool of extracting actual annotation examples that would be suitable for the annotation guideline or a tool for avoiding annotation errors.

As a remaining issue of the above contributions, when a classifier trained on the prepared data does not perform with sufficient accuracy, we need to consider how we increase training data effectively. That is, the issue we have to address is how we improve the classifier's accuracy for estimating values as few training data as possible. For this purpose, we can apply an active learning method that uses "pseudo-negative examples", which we also achieved as described in Appendix B (Takayama et al. 2009; Imamura, Takayama, et al. 2009). The key feature of the method is to suggest an appropriate example to add into training data with human annotation for improving classification effectiveness.

1.3 Structure of the Thesis

This section describes the construction of the rest of this thesis.

Chapter 2 investigates an effort to improve the identification of human values that are directly or indirectly invoked within the prepared statements of witnesses before legislative and regulatory bodies, focusing on one contentious political issue: net neutrality (Takayama et al. 2013). We automatically code, at the sentence level, human values that a writer has sought to reflect or appeal to when participating in the public debate, using supervised machine learning techniques trained on several thousand manually annotated sentences within 102 documents. To simulate an actual situation, we treat a quarter of the data as labeled for training and the remaining three quarters of the data as unlabeled for test. We find that augmenting the feature space using a combination of lexical and statistical co-occurrence evidence can yield an approximately six percent relative improvement in F_1 score compared to using an SVM classifier. However, it is revealed that the word categories (hypernyms) and the semantic class (synonyms) are not useful for detecting human values by the experiments using SVMs with augmented feature vectors.

Chapter 3 describes a probabilistic latent variable model that is designed to detect human values (Takayama et al. 2014). The proposed model treats the words in a sentence as having been chosen based on specific values; the values reflected by each sentence are then estimated by aggregating the values associated with each word. The model can determine the human values for the word in light of the influence of the previous word. This design choice was motivated by syntactic structures such as noun+noun, adjective+noun, and verb+adjective. The classifier based on the model was evaluated on the net neutrality corpus, achieving the highest reported classification effectiveness for this task. We also compared our proposed classifier with second human

annotator. As a result, the proposed classifier effectiveness is statistically significantly comparable to human annotators.

Chapter 4 introduces an automated method for facilitating social science research, specifically for content analysis (Fleischmann, Takayama et al. 2015). We explore an automatic learning method by simulated annealing, a meta heuristic algorithm, to extract a values-dictionary which identifies words that are strongly associated with sentences that human annotators coded as being related to specific values. This simple approach has an advantage of being able to obtain a values-dictionary and also yields nearly the same level of effectiveness as those achieved by an SVM.

Finally, Chapter 5 concludes this thesis and describes several future research directions to advance this study.

Chapter 2

Automatic Sentence Level Annotation using Augmented Feature Vectors

2.1 Introduction

Lexical features such as words and word stems have been shown to be a useful basis for studying affective dimensions such as sentiment or opinion when applied to first-person statements (Pang and Lee 2008; Liu 2011). Although sentiment analysis and opinion mining are useful in their own right, some social scientists have sought to look more deeply for factors that might help to explain, and perhaps ultimately to predict, sentiment and opinion (Fleischmann et al. 2012). In this chapter, we seek to advance one such line of work that is focused on automatic classification of human values such as *freedom* or *justice* to which writers of first person statements appeal.

In prior work, Ishita et al. (2010) have reported that lexical features can serve as a useful basis for classification of human values in the prepared statements of witnesses before legislative and regulatory hearings. The experiment in the prior work using *k*-NN (Nearest Neighbor) classifiers for 2,005 sentences over 28 documents obtained a macro-averaged F_1 of 0.48 for eight human values. To scale up social science research (Fleischmann et al. 2012), we need to improve our ability to effectively analyze larger data corpora. In this chapter, thus we employ a corpus containing 8,660 sentences in 102 documents over six values, which is four times larger than in the corpus used in the previous work (Ishita et al. 2010).

From a technical perspective, the use of lexical features alone has, however, done relatively poorly when applied to low-prevalence values categories such as *honor* (which was annotated by a human annotator as invoked in only 4% of the sentences in the test collection) that we use in our experiments. The reason for this problem seems to be that sentence-scale text classification necessarily results in feature sparsity (with sentences averaging just 10.3 word stems), and that the paucity of positive training examples for low-prevalence simply exacerbate that problem. An obvious approach would be to augment the feature set, an approach that is well known to be effective in text retrieval applications with short queries (so-called "query expansion") (Manning et al. 2008). The risk, of course, is that unconstrained feature augmentation can adversely affect precision by generating a substantial number of infelicitous matches. Threading this needle

between under- and over-augmentation therefore requires attention to constraining the search space.

In realistic situations, we usually have a small amount of labeled data annotated by humans and a large amount of related unlabeled data to be annotated. The role of our classifier here, of course, is assignment of human values as labels to unlabeled data. In many real-world situations, there would be a small, well-examined set of labeled data and larger collection of unlabeled data to be annotated. However, our corpus was relatively small (only 102 testimonies) and was already exhaustively annotated. Thus, we must create a virtual situation as similar as possible to evaluate the efficiency of our proposed classifier using augmented feature vectors. We therefore use a smaller portion of the corpus as labeled data for training and the remainder of the corpus as unlabeled data for test, and we also use the both data for extracting word associations to augmenting the feature vector, in order to simulate a realistic scenario.

This chapter also reports on an experiment comparing multiple human annotators and the classifier to explore the possibility of replacing human annotators with our classifier.

The remainder of this chapter is organized as follows. The next section describes human values inventory and our test collection. That is followed by our classifier design, results, and discussion in that order. The chapter concludes with a brief description of next steps.

2.2 Human Values Inventory and Test Collection

The section describes human values inventory and test collection we use in experiments.

2.2.1 The Meta-Inventory of Human Values

The value categories for this study were selected from the Meta-Inventory of Human Values (MIHV) (Cheng and Fleischmann 2010). The MIHV was developed to support content analysis of prepared testimonies presented at public hearings related to *the net neutrality* debate, building on earlier analysis of the role of values within a subset of this corpus (Cheng et al. 2012). The development process of the MIHV as follows.

Four rounds of refinement were conducted, seeking to optimize coverage of the values that writers drew on in this debate while maximizing inter-annotator agreement (Cheng 2012). Four rounds of annotation were conducted to refine the annotation guidelines. For each round, four documents were randomly selected from the corpus for annotation and two independent annotators. Cohen's Kappa (Cohen 1960) was used to characterize inter-annotator agreement, and Landis & Koch's guidelines (Landis and Koch 1977) were used to interpret the *Kappa* values, as is common in computational linguistics and other domains (Artstein and Poesio 2008).

Their initial annotation experience revealed poor inter-annotator agreement. After some iteration of annotation guidelines, they concluded that the Schwartz Values

Inventory (Schwartz 1994), which was developed through and for surveys, was not necessarily transferable to (manual or automatic) content analysis. To address this concern, Cheng and Fleischmann (2010) developed the MIHV by looking for commonalities among the full range of values inventories proposed towards categories that could be reliably inferred during annotation for content analysis. They selected a subset of their MIHV appropriate to the collection, iteratively coding a subset of o the collection and iteratively refined annotation guidelines using two annotators until inter-annotator agreement stabilized.

Among the 16 value categories in the MIHV, six value categories consistently achieved substantial agreement (κ=0.61 to 0.80) or moderate agreement (κ=0.41 to 0.60) throughout the four rounds of the annotation processes. These six human values were then used by the first annotator to annotate the entire corpus. Twenty documents from the corpus were annotated by a second annotator. The definitions of these six human values are in Appendix A.

2.2.2 Test Collection

The corpus for this study was created from written opening statements and testimonies (written statements) prepared for and delivered at public hearings held by the U.S. Congress (Senate and House) and the U.S. Federal Communications Commission (FCC) in which representatives of stakeholder groups offered advice to legislative and regulatory bodies on *net neutrality*. These were obtained from Lexis-Nexis Congressional web sites, and the FCC website. Each document was manually reviewed, and documents without any full-text content or with only slides were removed. The remaining 102 documents were used for the experiments reported in this thesis.

The key question in the net neutrality debate is whether the public interest is better served by nondiscriminatory access for all Internet traffic or by some set of reasonable policies for certain types of content or services. Their annotation task focused on the relationship between advocacy positions and detectable human values reflected by (or appealed by) written prepared statements. Manual annotation of a subset of this corpus has been used to discover relationships between values and sentiment (e.g., positive sentiment toward net neutrality was found to be correlated with the value *innovation*, and negative sentiment toward net neutrality was found to be correlated with the value *wealth* (Cheng et al. 2012). The ultimate goal of our work is to be able to replicate similar experiments at a larger scale.

Traditional paper-based annotations for values posed two challenges: (1) annotated passages could be of any length, and indeed both short (clause-scale) and long (paragraph-scale) passages were annotated; and (2) annotated passages often did overlap, indicating that evidence for multiple values was present in some places. Cheng et al. (2012) therefore elected to constrain the scope of each annotation to be a single sentence, but to allow more than one value per sentence. Clause annotations were extended to sentences, and passages that spanned sentences were accommodated by annotating

several consecutive sentences. Thus the sentence annotation was conducted to the development of the net neutrality corpus. This set up a well-structured sentence annotation task for supervised machine learning.

Sentence splitting for the test collection had been performed manually, and all 9,890 sentences in 102 documents were manually annotated by a social scientist. A total of 7,901 sentences invoked at least one value (minimum 1, median 1, mean 1.64, maximum 5). No value categories were assigned to the remaining 1,989 sentences, 340 of which were annotated as section headings. The average sentence length is 16.5 words.

Table 2-1 shows examples of annotated sentences. We subsequently removed sentences whose boundaries disagreed with those of TreeTagger (Schmid 1994). We removed sentences annotated as section headings then 8,713 sentences remained. Then we removed the sentences that after removing words in the SMART stopword list (Salton and McGill 1988) contained more than 40 words. Finally we also removed null sentences after eliminating stop words, leaving 8,660 sentences. The remaining 8,660 sentences were then stemmed by the Porter stemmer (Porter 1980). The average sentence length is 10.3 word stems.

Table 2-2 shows the distribution across the six values for original net neutrality corpus and the corpus actually used in experiments. A total of 1,545 sentences were annotated as containing *no value* in the corpus revised.

A second annotator independently had annotated 20 of the prepared statements (containing 2,430 sentences, after the same filtering process was applied). Table 2-2 also shows Cohen's kappa, a chance-corrected measure of inter-annotator agreement (Artstein and Poesio 2008; Cohen 1960; Landis and Koch 1977) for those 20 documents.

Table 2-1. Examples of human values annotation.

Values	Sentence
freedom, social order	Consumers are entitled to access the lawful Internet content of their choice.
honor	I am one of the network engineers involved for many years in designing, implementing and standardizing the software protocols that underpin the Internet.
innovation, freedom	Part of the reason why the Internet is such a creative forum for new ideas is that there are very few barriers to using the Internet to deliver products, information and services.
justice	Under these circumstances, requiring those most responsible for congestion to bear a greater percentage of the costs would be both good network management and fair from a consumer standpoint.
social order	The Commission, under Title I of the Communications Act, has the ability to adopt and enforce the net neutrality principles it announced in the Internet Policy Statement.
wealth	Private investors will fund the construction of a broadband network only if there is a reasonable expectation that the company making that investment will recover the cost of its investment, including a competitive return on capital.

Table 2-2. Inter-annotator agreement and prevalence.

Value	original			used (8,860 sentences)		
	Kappa	# docs	# sentences	*Kappa*	# docs	# sentences
wealth	0.629	102	3,563	0.621	102	3,156
social order	0.683	102	2,859	0.688	102	2,503
justice	0.420	99	2,641	0.423	99	2,267
freedom	0.620	101	2,431	0.628	101	2,155
innovation	0.715	94	1,147	0.714	94	1,018
honor	0.430	80	352	0.437	80	317

2.3 Augmented Feature Vectors

This section describes our approach to automating annotation of human values. We adopt SVMs (Support Vector Machines) as our classifiers, which are among the most effective known approaches for document categorization (Sebastiani 2002). An SVM is a vector space machine learning method which can work effectively in a high dimensional input space, so there is no reason not to consider expanding the *baseline vectors* computed directly from term occurrence in each sentence to partially model the background knowledge that a human reader brings to the interpretation of a sentence. From the perspective of the SVM classifier, this approach serves to help mitigate sparsity in the feature space. We consider two types of expansion strategies.

Our first type of strategy relies on statistically associated terms as a basis for expanding the feature set for each sentence. Specifically, we model the semantic relatedness between words by same-sentence co-occurrence statistics in some representative corpus (in our case, the full corpus being classified). We tried two specific measures to calculate term association: (1) CP (Conditional Probability), and (2) PMI (Pointwise Mutual Information) (Church and Hanks 1990; Hindle 1990). Both are unsupervised, not requiring any human annotation.

$$CP(w_j \mid w_i) = \frac{P(w_i, w_j)}{P(w_i)} = \frac{freq(w_i, w_j)/N}{freq(w_i)/N}, \tag{1}$$

$$PMI(w_i, w_j) = \log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)} = \frac{freq(w_i, w_j)/N}{\left(freq(w_i)/N\right)\left(freq(w_j)/N\right)}, \tag{2}$$

where $freq(w_i)$ and $freq(w_j)$ are term occurrence counts, $freq(w_i, w_j)$ is the term pair co-occurrence count, and N is total number of sentences in the corpus from which we learn the association statistics. To be used for expansion, we require that $freq(w_i)>\theta_1$, $freq(w_j)>\theta_1$ and $freq(w_i, w_j)>\theta_2$ (for our experiments we use $\theta_1=\theta_2=10$). For PMI, we accept all expansion terms with positive PMI values. For CP, we accept all expansion terms with the values between 0 to 1. Because we wish to learn the association statistics from a closely related corpus, we actually learn our term association statistics from the same collection that we use for evaluation. Both approaches are unsupervised (requiring no annotations), so using the evaluation corpus itself is practical, and representative of what could be done in a real applications.

Our second type of expansion strategy relies on synonymy ("*syn*") or (one-step immediate) hypernymy ("*hyp*") relations that are encoded in a lexicon. Specifically, we

use all matches to each lemmatized word that we find in the noun, verb, and adjective hierarchies in a thesaurus. Because we expect the thesaurus to have limited coverage of domain-specific terminology in any specific domain, we expect this approach to be most useful for general terminology.

Figure 2-1 outlines our algorithm, with term association statistics (CP and PMI), and term relatedness (synonymy and hypernymy) found in steps 01 and 02, respectively. In step 03, the training data Tr_i for fold i of the cross-validation consists of baseline binary vectors after stemming using a stemmer (i.e., word stem id's with value 1 occur in the corresponding sentence; others have value 0). If we freely expand the term vectors based on word association or on the lexicon in steps 07 or 08, respectively, the components of the augmented vectors would become denser (i.e., more 1's from lexicon and real values for association), but less precise. We limit the loss of precision by first choosing, in step 06, which *base stems* to use as a basis for expansion. To select this smaller *base stems* vector, in step 04 we train a classifier $\Gamma^{(v)}_{\text{base}}$ for each label to identify the stems whose presence is positively correlated with the presence of that label in the training set Tr_i, repeating the process for each category and within each category for each fold i. If the correlation is positive (i.e., greater than zero) in step 06, the stem is treated as a *base stem* for expansion.

01: Create associative word dictionary using distributional similarities by (1) Conditional Probability and (2) PMI.

02: Create (a) synonym and (b) hypernym dictionary by consulting a hand-crafted lexicon.

03: Prepare cross-validation data set, where $Tr_1, ..., Tr_N$: training data, $Te_1, ..., Te_N$: test data, and both Tr_i and Te_i ($1 \leq i \leq N$) consist of word id's with score 1, that occurred in each sentence. (The suffix N represents N-fold cross-validation.)

04: Construct classifier $\Gamma^{(v)}_{\mathrm{base}}$ by learning the training data $Tr_1, ..., Tr_N$ for each value (v).

05: Classify $Te_1, ..., Te_N$ by the classifier $\Gamma^{(v)}_{\mathrm{base}}$, and evaluate the result as the baseline effectiveness.

06: Choose the *base stems(i)* for expanding the feature vectors, which contribute to classification for each value in each training data Tr_i.

07: Expand feature vectors as Pr_i, Pe_i with PMI score and Cr_i, Ce_i with Conditional Probability using associative word dictionary, for each *base stem (i)*.

08: Expand feature vectors as Sr_i, Se_i for synonyms and Hr_i, He_i for hypernyms using synonym and hypernym dictionary, for each *base stem (i)*.

09: Construct the augmented vectors for the both training data $ATr_1, ..., ATr_N$ and test data $ATe_1, ..., ATe_N$.

$$ATr_i = Tr_i \quad [+ Pr_i] \; [+Cr_i] \; [+Sr_i] \; [+Hr_i],$$

$$ATe_i = Te_i \quad [+ Pe_i] \; [+Ce_i] \; [+Se_i] \; [+He_i],$$

where + represents the vector concatenation operator and [ ] represents optional.

10: Construct classifier $\Gamma^{(v)}_{\mathrm{modified}}$ by learning the augmented training data $ATr_1, ..., ATr_N$.

11: Classify the augmented test $ATe_1, ..., ATe_N$ by $\Gamma^{(v)}_{\mathrm{modified}}$ and evaluate the result.

Figure 2-1. Classification with augmented feature vectors.

2.4 Experiments

In this section we report results for classifier selection and for classification, with and without expansion.

2.4.1 Preliminary Result and Classifier Design

Before constructing of word vectors, we apply the following preprocessing steps.

(1) Lemmatization using TreeTagger 3.2,[1] to normalize each word to its corresponding WordNet[2] (Miller 1995) root form.
(2) Stopword removal using the SMART stopword list,[3] adapted for TreeTagger's output.
(3) Stemming by Porter's stemmer (Porter 1980).

Before focusing on SVM results, we conducted a preliminary experiment comparing *k*-Nearest Neighbor (*k*-NN), naive Bayes (NB) and SVM classifiers. We used the University of Waikato's Weka toolkit[4] for *k*-NN (with *k*=1) and NB, and throughout this chapter we use TinySVM[5] (with a second-degree polynomial kernel) as our SVM classifier. Table 2-3 shows the results for 102-fold document cross-validation (i.e., the average over 102 classifiers, each trained on some set of 101 documents and tested on the one remaining held out document). For example, a sentence in a training document that was annotated with *freedom* and *innovation* would be a positive training example for each of those categories and a negative training instance for all other categories. Sentences annotated with no value categories are used as negative training examples for all categories. The SVM yielded the best results among the three classifiers by precision, recall, and F_1, so we focus on SVM classifiers for the remainder of this chapter.

Table 2-3. Classifier selection (102-fold document cross-validation).

Classifier	Precision	Recall	F_1
k-NN (*k* = 1)*	0.6526	0.3737	0.4658
naive Bayes*	0.5996	0.6737	0.6338
SVM (2nd poly)	0.7730	0.6510	0.7068

* The scores by Weka here used all word stems, while those in (Takayama 2013) used only 1,000 stems.

Each document in our corpus is the prepared testimony of a witness before a regulatory or legislative hearing, and human annotation was done one document at a time. Thus in next three subsections, we divide the 102 documents at document

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[2] http://wordnet.princeton.edu/wordnet/

[3] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[4] http://www.cs.waikato.ac.nz/ml/weka/

[5] http://chasen.org/~taku/software/TinySVM/

boundaries in a way that uses approximately 25% of the sentences for training and the remaining 75% for testing. This represents a realistic situation in which a user might reasonably completely annotate a few dozen documents as they work or an insightful and reliably usable coding frame and then uses an automated system to annotate dozens or even hundreds more. In step 03 of Figure 2-1, we repeat this 25% / 75% split 102 times, each time anchoring a different document as the center of the evaluation subset for that 102-fold cross-validation.

2.4.2 Comparison with Human Annotation

To see how our SVM classifier compares with human annotator agreement on a per-category basis, we need to test on a single set of documents that have been multiply annotated. A total of 20 documents were therefore annotated for this purpose by a second annotator. We treat the first annotator's annotations of those 20 documents as correct, and compute classifier effectiveness measures as if our additional annotator were a classifier, as the left columns for each evaluation measure in Table 2-4. Comparable results for the baseline classifier, tested on the same 20 documents and trained on the remaining 82 documents, are shown at the right columns for each evaluation measure in Table 2-4. As can be seen, our baseline classifier does about as well as a human second annotator did on *social order* and *freedom*, and it actually does a bit better than our second annotator did on *justice*!

Under comparable conditions, but using expansion PMI' + syn', both with our *base word* constraint), we get about the same average F_1 (0.6991). From this we conclude that once we have enough training data, expansion is of little help overall (although we do see a 4% relative gain in *honor* from expansion, perhaps because *honor* has the fewest positive training examples). We compute relative improvement as $(b - a) / a$, where a and b are the two efficiency values being compared.

Table 2-4. Human "classifier" and SVM effectiveness (20 documents).

	Precision		Recall		F_1	
Value	human	SVM	human	SVM	human	SVM
wealth	0.7345	0.8160	0.8711	0.6536	0.7970	0.7258
social order	0.7751	0.7828	0.7588	0.7468	0.7669	0.7644
justice	0.6635	0.6518	0.4638	0.5691	0.5460	0.6076
freedom	0.6810	0.7362	0.7682	0.7048	0.7220	0.7201
innovation	0.7644	0.7800	0.7197	0.6393	0.7414	0.7027
honor	0.3950	0.4800	0.5529	0.1412	0.4608	0.2182
average	0.7117	0.7550	0.7320	0.6505	0.7217	0.6988

2.4.3 Overall Effect of Expansion

Table 2-5 shows results for the unexpanded baseline, and for our several variants of expansion, as averages over the six human values for precision, recall, and F_1. The symbol "+" represents the vector concatenation operator and the symbol ' (for "constrained") means that the expansion is constrained to be based only on the *base stems* chosen in step 06. As can be seen, the constraint is helpful when lexicon-based expansion is used, but it is not necessary (and indeed it seems harmful) when CP or PMI association scores are used (because CP and PMI already include a selection threshold).

Table 2-5. Average (25% train:75% test, 102-document cross-validation).

Method	Precision	Recall	F_1
(0) : baseline	0.7515	0.5108	0.6082
(0)+(1) : CP	0.7249	0.5744	0.6410
(0)+(1)' : CP'	0.7487	0.5158	0.6108
(0)+(2) : PMI	0.6760	0.5956	0.6332
(0)+(2)' : PMI'	0.7478	0.5137	0.6090
(0)+(a) : syn	0.7067	0.4676	0.5628
(0)+(a)' : syn'	0.7487	0.5159	0.6109
(0)+(b) : hyp	0.6906	0.5068	0.5846
(0)+(b)' : hyp'	0.7485	0.5154	0.6105
(0)+(b)'+(c)' : syn' + hyp'	**0.7713**	0.5153	0.6103
(0)+(1) +(a)': CP + syn'	0.7278	0.5761	**0.6432**
(0)+(2) +(a)': PMI + syn'	0.6756	**0.6005**	0.6359

As comparing Tables 2-3 and 2-5 shows, F_1 declines by about 0.1 absolute when trained with 25% rather than 99% of the documents (compare 0.7068 with 0.6082). Comparing the best results in Table 2-5 with the baseline indicates that augmenting feature vectors using both CP and syn' recovers some of that loss, yielding a 0.035 absolute improvement in F_1 over the baseline that uses only lexical features (compare 0.6432). From this we conclude that expansion is most useful when only a limited number of training documents can be annotated (as is the case in many practical applications).

2.4.4 Per-Category Analysis

Averages can hide important details, so we also report results for each of our six human value categories. Table 2-6, corresponds to the first line in Table 2-5; Table 2-7 corresponds to the second to last line in that table (expansion using CP+syn', which yields gives the best average F_1). Table 2-8 shows the relative improvements in F_1, which average about 6% ((0.6432-0.6082)/0.6082). Again, we see the largest improvement for *honor*, for which the fewest positive training examples are available.

Table 2-6. Baseline (25% train:75% test, 102-document cross-validation).

Value	Precision	Recall	F_1
wealth	0.7699	0.5529	0.6436
social order	0.8203	0.6243	0.7090
justice	0.6636	0.3796	0.4829
freedom	0.7025	0.5377	0.6092
innovation	0.8308	0.4694	0.5998
honor	0.3490	0.07187	0.1192
average	0.7515	0.5108	0.6082

Table 2-7. CP+syn' Classifier (same condition as Table 2-6).

Value	Precision	Recall	F_1
wealth	0.7547	0.6247	0.6835
social order	0.8014	0.7019	0.7483
justice	0.6368	0.4693	0.5404
freedom	0.6790	0.5762	0.6233
innovation	0.7843	0.5024	0.6125
honor	0.3511	0.0884	0.1413
average	0.7278	0.5761	0.6432

Table 2-8. Relative F_1 improvement (from Table 2-6 to 2-7).

Value	*wealth*	*social order*	*justice*	*freedom*	*innovation*	*honor*
F_1 improvement	6.2%	5.6%	11.9%	2.3%	2.1%	18.5%

2.4.5 Sentence Cross Validation

In prior work, Ishita (2010) has reported results for 10-fold cross-validation, using randomly selected sentences. Table 2-9 shows results for that design (with our present values categories; in our earlier work we had used a different values inventory).

Table 2-9. Classifier effectiveness (10-fold sentence cross-validation).

Method	Precision	Recall	F_1
(0) : baseline	0.7868	0.6672	0.7221
(0)+(1): CP	**0.8077**	0.6547	0.7232
(0)+(1)': CP'	0.7936	0.6674	0.7251
(0)+(2): PMI	0.7996	0.6527	0.7187
(0)+(2)': PMI'	0.7935	0.6672	0.7249
(0)+(a)': syn'	0.7840	0.6701	0.7226
(0)+(b)': hyp'	0.7838	0.6704	0.7227
(0)+(2)'+(a)': PMI' + syn'	0.7963	**0.6704**	**0.7279**

As can be seen, because random sentence selection can divide sentences from the same document between the training and test sets of the same fold, the baseline results exceed that of even 102-fold document cross-validation (i.e., better baseline F_1, even with a bit less training data). Moreover, we see a somewhat different pattern of comparisons (e.g., now the base stems constraint helps PMI rather than hurting it). Because randomly selecting sentences does not model the real annotation task as well as selecting entire documents would, we caution against using 10-fold sentence cross-validation for these types of experiments.

2.5 Summary

In this chapter, we have applied SVMs with augmented feature vectors to identify human values for sentences to automate content analysis in social science. The key issue that we have addressed is conquering sparsity. We proposed the combination of evidence from statistical term associations and lexical evidence for hypernyms and synonyms as elements of feature vectors. We have improved substantially over previously reported results (Ishita et al. 2010) by using annotations based on a new human values inventory that are well matched to our task. We have adopted a more realistic (and more conservative) document-selection approach to cross-validation, and we have demonstrated that improvements in the effectiveness of sentence classification can be achieved using expansion.

We now have some degree of confidence that we might reasonably apply our classifiers in support of some types of social science at a far larger scale than would be possible using human annotations alone, which could help us to find interesting signals within the larger scale information (Fleischmann et al. 2012).

However, by the experiments in this chapter, we found that the use of word categories (hypernyms) and word semantic classes (synonyms) fails to work properly for estimating human values using SVMs with augmented feature vectors, while the word semantic classes are usually effective for word sense disambiguation or syntactic disambiguation. This reveals that class-based generalization of words might prevent capturing sensitivity of human values. The fact led us the next direction of consideration of another method that assigns the values directly to words or words pairs.

Chapter 3

A Word-Scale Probabilistic Latent Variable Model for Detecting Human Values

3.1 Introduction

Social scientists have long found it useful to consider human values as latent variables that have explanatory value for the choices that people make (Verplanken and Holland 2002). For example, someone who values *innovation* over *wealth* might advocate open-source over proprietary software, while someone who values *freedom* over *social order* might resist efforts for gun registration. We can think of values as influencing not only how people form their own opinions, but also as undergirding how people seek to influence the opinions of others. In this chapter, we focus on automatic detection of human values reflected in texts written by advocates of specific policy positions. We take a step in that direction by evaluating automated classification of human values.

Several inventories of human values are used in social science research (e.g., Friedman et al. 2006; Kahle et al. 1988; Kluckhohn 1951; Rokeach 1973; Schwartz 1994). Integrating key components of these studies, we adopted Cheng and Fleischmann's (2010) human value definition, that is, "values serve as guiding principles of what people consider important in life." We also base our work on the Meta-Inventory of Human Values (MIHV), which was developed by Cheng and Fleischmann specifically for the test collection that we use by selecting values specific to the debate at issue and by iteratively refining annotation guidelines (Fleischmann et al. 2009; Cheng et al. 2012; Fleischmann 2014). Our results, generated using a redistributable collection containing 102 documents with zero or more of six sentence-level human values annotations, indicate that high precision (near 0.8) can be reliably achieved for frequently invoked values with a useful degree of recall (0.55-0.82, except for *honor*).

We achieved statistically significant classification effectiveness over existing baselines for this task using a new probabilistic latent variable model in which we first infer the association between human values and individual word-level human values as latent variables, and then we aggregate those results over all words in a sentence. The structure of our model allows us to model the potential effect of the preceding word, which proves to be useful. Moreover, analysis of 20 dual-annotated documents indicate that with about 80 training documents our automated technique is able to achieve

results that are nearly as accurate as those obtained by an independent human annotator as a pseudo-classifier.

The remainder of this chapter is organized as follows. In Section 3.2, we describe related work on human value research and on classification methods. Section 3.3 then looks back the test collection that we have used. Section 3.4 describes our approach to detect human values and Section 3.5 describes our proposed latent value model. Section 3.6 represents our results and Section 3.7 summarizes the chapter.

3.2 Related Work

Content analysis is one of approaches to detect human values (Fleischmann et al. 2012). The key idea in content analysis is for the social science researchers to personally examine naturally occurring content and to assign codes to that content that reflect their interpretation of that content using some pre-existing coding scheme. Subsequent statistical analysis is then done on the assigned codes rather than on the content. Hsieh and Shannon (2005) refer to this combination of human interpretation and an existing coding scheme as a "directed approach". One of the limiting factors the directed approach is that the annotation costs scale linearly with the size of the collection. Early in the annotation process, personal involvement of the researcher is important because the theory on which any pre-existing coding scheme is built may need to be adapted for reflecting the unique characteristics of a collection on which social scientists wish to focus. Our automated techniques are intended only for the part of the process when coding guidelines have stabilized and a substantial amount of annotated data is available.

After we obtained sufficient annotated data, we could automate annotation process using text classifiers (Templeton et al. 2011) trained with that data. We are not the first to explore the automated annotation of human values for social science research. For example, Bengston et al. (2004) used dictionary-based computer aided content analysis to identify how values about forestry have shifted from anthropocentric values to biocentric values over the period 1980 through 2002.

We first compared the effectiveness of a wide range of classifiers available within Weka (Hall et al. 2009), and we found that Support Vector Machines (SVMs) (Joachims 1998, 2002) performed best. Therefore, we compare our proposed method to SVMs using bag-of-words and bigram features in Section 3.6. Because we introduce a latent variable model, supervised Latent Dirichlet Allocation (sLDA) offers another appropriate baseline (Blei and Mcauliffe 2007). Essentially, sLDA is an extension of LDA (Blei et al. 2003) in which the process of constructing the probabilistic latent variable model is influenced by the known association of words with labels in a set of training documents. sLDA based on generalized linear models is a general framework to model the documents and the responses. Our proposed probabilistic latent variable model also captures the relationships between the sentences and values. Thus, we compare our method with sLDA in Section 3.6.

Labeled LDA (L-LDA) (Ramage et al. 2009) is an extension of both LDA and multinomial naive Bayes. L-LDA is a generative model for multi-labeled document collections and only one label is assigned to a word in a document. L-LDA might be well performed for document collections, however, our task is sentence-level detection of human values. Therefore, we will design an original discriminative model that assigns to multiple-human values to a word under the constraint of the sentence-level values.

Griffiths et al. (2005) found modeling sequential dependencies between word classes to be helpful. Sequential dependencies between the words themselves can also be useful, but sparsity risks must be managed. With this in mind, we model sequential word dependencies with the label(s) assigned to one word stem depending in part on the label(s) assigned to only the previous word stem.

The structure of our problem resembles that of sentiment classification, which has been extensively researched (Pang and Lee 2008; Liu 2011). An important difference is that our classification of human values is most naturally cast as a multi-category multi-label classification, whereas sentiment analysis is typically modeled as single-category classification. Importantly, human values can help to explain sentiment, given their explanatory power in relation to attitudes and behavior (Wiebe 1994). What distinguishes our work is our focus on human values with a redistributable test collection and our modeling of relation between sentence-level and word-level values, and sequential dependencies among words in a sentence.

3.3 A Test Collection for Human Values

The section looks back human values and test collection we use in experiments.

As described in Section 2.2, the test collection we used in experiments was originally developed by Cheng et al. (2012). The collection includes 102 written prepared statements ("testimonies") from public hearings held by the U.S. Congress and Federal Communications Commission (FCC) on net neutrality. Table 3-1 shows some example sentences annotated with some of six human values: *freedom*, *honor*, *innovation*, *justice*, *social order*, *wealth*. Actual test corpus contains 8,660 sentences in 102 documents were then stemmed by the Porter stemmer (Porter 1980). The average sentence length of our corpus is only 10.3 word stems.

Table 3-1. Examples of values annotation.

Values	Sentence
freedom, social order, honor	This Committee has a long history of overseeing developments in communications industries and the Internet, and you have diligently promoted policies to ensure competition in these markets.
innovation, wealth	Akamai thus represents a creative way to use server hardware as a substitute for network hardware.
justice	Survival of the Internet requires that Internet Access Providers continue to take a proper, transparent role as participants in the Internet.

3.4 Approach for Detecting Values

In order to detect human values, we have to take into account how the values are reflected in text. Surface language expressions for human values are different from those for most other subject classification problems. In using subject classification to classify a theme of the document, the themes are often directly represented by language expressions, typically by words that occur in the documents (Sebastiani 2002). In the case of the human value classification, while a value may be indicated by a specific word in some cases, in many cases the value may be invoked somewhat more indirectly using situation-specific terminology.

In a preliminary analysis of the corpus that we use in this chapter, we found the following cases:

(1) A word represents value(s).

The word in a sentence represents the certain values. For example, the word "freedom" in sentence (a) expresses the value *freedom*, the word "protect" in sentence (b) expressed the value *social order*, and the word "winner" in sentence (c) expressed the both of *justice* and *wealth*. As shown in (a), value names themselves are usually good cue words for the values.

(a) "This preserves consumers' freedom to go where they want, use the lawful services they want, and read and say what they want online."

(b) "Protecting customers and delivering a good Internet experience is not limited to curtailing spam or thwarting identity theft, for example."

(c) "Consumers, not network operators, must be allowed to continue to choose winners and losers in the content and applications marketplace."

(2) A pair of words represents value(s).

The following sentence (d) has the value *innovation*, but the sentence (e) does not. The word pair "good idea" (adjective+noun) provides the value *innovation* ("idea" means *a suggestion for possible course of action*), but the word "idea" in the sentence (e) does not specify any value ("idea" means just *a thought*).

(d) "Make sure there is always a fertile place for all of our good ideas to flourish."

(e) "That was, I believe, the first time that idea had been presented to this Committee."

(3) A whole sentence represents value(s).

The annotator determined that the following sentence (f) invokes the value of *honor* based on its statement.

(f) *"I am one of the network engineers involved for many years in designing, implementing and standardizing the software protocols that underpin the Internet."*

(4) Contextual information is required to infer value(s).

The following sentences (h) and (i) are annotated based on context. Sentence (h) has *honor* because of the previous sentence (g) which has the values *honor*, *innovation*, and *wealth*. Sentence (i) has *freedom* and *wealth* with influence by the next sentence (j) (which also has the values *freedom* and *wealth*).

(g) *"This is an extraordinarily positive development for the nation's economy . . . for our global competitiveness . . . and for the next wave of broadband-driven investment and innovation."*

(h) *"How do we continue this progress?"*

(i) *"First and foremost, by recognizing that this market is contestable to all who wish to invest."*

(j) *"This is plainly evidenced by the growing array of companies doing just that in the marketplace today ... cable ... phone ... satellite power ... municipality ... WiFi ... WiMax ... Google and more ... all investing in what is increasingly a free-for-all for consumers' broadband business."*

From the above actual examples in the corpus, we can see that the human values are expressed in variety of forms and multiple values are assigned to a sentence.

Among several approaches to estimate the presence of a category from text, typical basic methods are naive Bayes, *k* nearest neighbors (*k*-NN), and SVMs (Manning et al.

2008). Ishita et al. (2010) and Takayama et al. (2013) adopted these methods to detect human values, however, the results showed that these methods alone are not sufficient. One reason is that human values cannot be represented by simple functions such that summation of factors of words in a sentence, contributing to each human value. These function cannot be capture some specific words play a determining role to detect certain values. Based on the above considerations, we design our model to first infer the word-level human values corresponding to each word in a sentence as latent variables, and then aggregate them by logical bitwise OR (see in Section 3.5.1) to estimate the sentence-level human values.

Another characteristic of language expressing human values is that multiple values can be expressed by a single sentence. There are the cases in which one word reflects multiple values, as example (c) above illustrates, and multiple words with values can appear in a sentence. As an example of that, the sentence "Part of the reason why the Internet is such a creative forum for new ideas is that there are very few barriers to using the Internet to deliver products, information and services." has the value *innovation* based on the word "creative" and the word pair "new ideas" (adjective+noun); and *freedom* based on the word "barrier."

The above examples (d) and (e) in the case (2) suggest that word sense disambiguation directed by syntax is required to detect correct human values for word-level. Among several syntax patterns, we focus in our work on two-word collocations, modeling the value of the word in a way that can be influenced by the previous word because this covers many typical and frequent syntax patterns, as the above examples show.

In this chapter, we model cases (1) and (2) above in the next section, that provide an adequate coverage of major cases, in anticipating that the above cases (3) and (4) are minor. We expect our design choice is effective for these cases, and in our future research, we could perhaps further extend our model to represent whole-sentence meanings and long-distance context in more nuanced ways.

3.5 Probabilistic Latent Value Model

In this section, we propose a new method for detecting human values by using a statistical language model we call our *Latent Value Model* (*LVM*, for short), that estimates the posterior probability of sentence **w** having values v using Gibbs sampling in a Markov Chain Monte Carlo framework (Griffiths and Steyvers 2004). In order to investigate relationships between words for detecting values as discussed in section 3.4, we take the effect of the preceding word into account in our *LVM*.

3.5.1 Preparation and Notation

A sentence **w** is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, ..., w_N)$, where w_n is the n-th word in the sequence. The sentence **w** has *sentence-level values* v, where $v \in \{0,1\}^6 =$ {000000, 000001, ..., 111111}. Each bit in the bit sequence pattern represents one of the

six values. The pattern 000000 means the corresponding sentence does not have any values.

We introduce latent variables **x** into the model to represent the value(s) associated with each word in the sentence. If the word w_n has a value x, the sentence **w** also has the value x. On the other hand, if no word w_n in a sentence **w** has value x, then the sentence **w** does not have value x. In addition, we assume that each word in a sentence has at most two values. The sequence of the values corresponding to the sentence **w** is denoted by $\mathbf{x} = (x_1, x_2, ..., x_N)$, where x_n is the *word-level value(s)* of the word w_n.

We restrict x_n to be an element in χ, where χ ={000000, 000001, 000010, 000011, 000100, ... , 110000}. The cardinality of χ is 22. We denote 000000 as μ_0, 000001 as μ_1, ..., and 110000 as μ_{21}, for convenience of notation. Restricting the number of values with which a word can be associated limits sparsity. Whether an at-most-two model is a good choice is an empirical question. In preliminary experiments, single-value models perform poorly and three values models show no further improvement.

The sentence-level values v are the result of logical bitwise OR operation $\oplus$ for all x_n ($1 \leq n \leq N$). The sequence of word-level value(s), therefore, is restricted to the following $\chi^N(v)$, when the sentence-level values v are given.

$$\chi^N(v) = \{(x_1, x_2, \cdots, x_N) \in \chi^N \mid x_1 \oplus x_2 \oplus \cdots \oplus x_N = v\}.$$

For example, in the sentence "A decision by a broadband Internet access provider to block specific content, so long as it is not motivated by anticompetitive objectives, is likely to be a form of protected speech for the provider" the word "anticompetitive" evokes the values *wealth* and *freedom*.

We also introduce another type of latent variable $\mathbf{y} = (y_1, y_2, ..., y_N)$ into the model. The *context indicator* y_n expresses whether the previous word w_{n-1} influences the value of w_n or not. When the values associated with word w_n are subject to the influence of the previous word w_{n-1}, the context indicator y_n takes the value 1. This design choice is motivated by syntactic structures such as noun+noun and adjective+noun, or semantic disambiguation associated with verb+noun and verb+adjective. Otherwise, y_n takes 0 (The values associated with w_n are determined by only w_n itself).

3.5.2 Model and Estimation of Values

(1) Model

For the word sequence (w_{n-1}, w_n), the context indicator y_n follows a Bernoulli distribution $Bern(\theta_1^{(w_{n-1}, w_n)})$ and its parameter $\theta_1^{(w_{n-1}, w_n)}$ follows a Beta distribution with the meta-parameters (a_0, a_1). When y_n takes the value 0, the values associated with the word w_n follow a multinomial distribution $Multi(\phi_0^{(w_n)}, \phi_1^{(w_n)}, ..., \phi_{21}^{(w_n)})$ and its parameters $\phi_{0:21}^{(w_n)}$ follow a Dirichlet distribution with the meta-parameters (b_0, b_1, ..., b_{21}). Hereafter, we use $\phi_{0:21}^{(w_n)}$

and $\beta_{0:21}$ for notational convention. When the context indicator y_n takes 1, the values associated with the word w_n follow a multinomial distribution $Multi\,(\phi_0^{(w_{n-1},w_n)},\phi_1^{(w_{n-1},w_n)},\cdots,\phi_{21}^{(w_{n-1},w_n)})$ and its parameters $\phi_{0:21}^{(w_{n-1},w_n)}$ follow a Dirichlet distribution with the meta-parameters $\beta_{0:21}$. The generative process for the sequence of sentence-level value patterns $\mathbf{x} = (x_1, x_2, ..., x_N)$, the sequence of context indicators $\mathbf{y} = (y_1, y_2, ..., y_N)$ and the sentence-level human value(s) v for a sentence $\mathbf{w} = (w_1, w_2, ..., w_N)$ in our proposed *LVM* is as follows in Figure 3-1. In Figure 3-1, priors $\theta_u^{(w_{n-1},w_n)}$ are called *contextual affinities*, $\phi_{0:21}^{(w_n)}$ and $\phi_{0:21}^{(w_{n-1},w_n)}$ are *word-level associations*.

Our *LVM* is represented as a graphical model using conditioning with gates (Minka 2008) in Figure 3-2. The outer plate represents sentences, while the inner plate represents generation of word-level values from a pair of the words or a single word by its context. The dotted box inside the inner plate shows the determination of previous word's influence depending on the context indicator y. The sentence-level value v is an aggregation of word-level values $\mathbf{x}$ for the corresponding sentence $\mathbf{w}$. In Figure 3-2, L is the number of distinct (w_{n-1}, w_n), W is the number of vocabulary and K is fixed at 22.

That is, our proposed model can be represented by the following equation (1).

$$P(\mathbf{x},\mathbf{y}\mid\mathbf{w},\theta,\phi) = \prod_{n=1}^{N} P(y_n \mid w_{n-1}w_n,\theta)\times P(x_n \mid y_n, w_{n-1}w_n,\phi), \tag{1}$$

where w_0 is the special symbol ($) expressing the sentence head, and y_1 is always 0. The probabilities $P(y_n \mid w_{n-1}, w_n, \theta)$ and $P(x_n \mid y_n, w_{n-1}, w_n, \phi)$ are defined as follows:

$$P(y\mid a,b,\theta)=\begin{cases}\theta_0^{(a,b)} & ;\ y=0\\ \theta_1^{(a,b)} & ;\ y=1\end{cases},\quad P(x=\mu_j\mid y,a,b,\phi)=\begin{cases}\phi_j^{(b)} & ;\ y=0\\ \phi_j^{(a,b)} & ;\ y=1\end{cases}. \tag{2}$$

For simplifying notation, the symbol a represents the word w_{n-1}, and the symbol b represents w_n, the previous word of w_n in the equation (2), and the same style notation shall apply hereafter. The constant μ_j in equation (2) is the j-th possible word-level value(s) pattern as described in section 3.5.1.

We assume the following properties about the relation between words and their values:

(a) Most words do not have any values,
(b) For most two-word sequences, the values associated with the second word are probabilistically determined by that second word alone, without influence from the previous word.

foreach $n = 1, 2, ..., N$ **do**

(i) **draw** *context indicator* y_n:

$$y_n \mid \theta_1^{(w_{n-1}, w_n)} \sim Bern\ (\theta_1^{(w_{n-1}, w_n)}).$$

(ii) **draw** *word-level value(s)* x_n :

if $y_n = 1$ **then**

$$x_n \mid w_{n-1}, w_n, \phi \sim Multi\ (\phi_0^{(w_{n-1}, w_n)}, \phi_1^{(w_{n-1}, w_n)}, \cdots, \phi_{21}^{(w_{n-1}, w_n)}),$$

else $x_n \mid w_n, \phi \sim Multi\ (\phi_0^{(w_n)}, \phi_1^{(w_n)}, \cdots, \phi_{21}^{(w_n)})$.

sentence-level value(s) become:

$$v = x_1 \oplus x_2 \oplus \cdots \oplus x_N .$$

Figure 3-1. Generative Process of *LVM*.

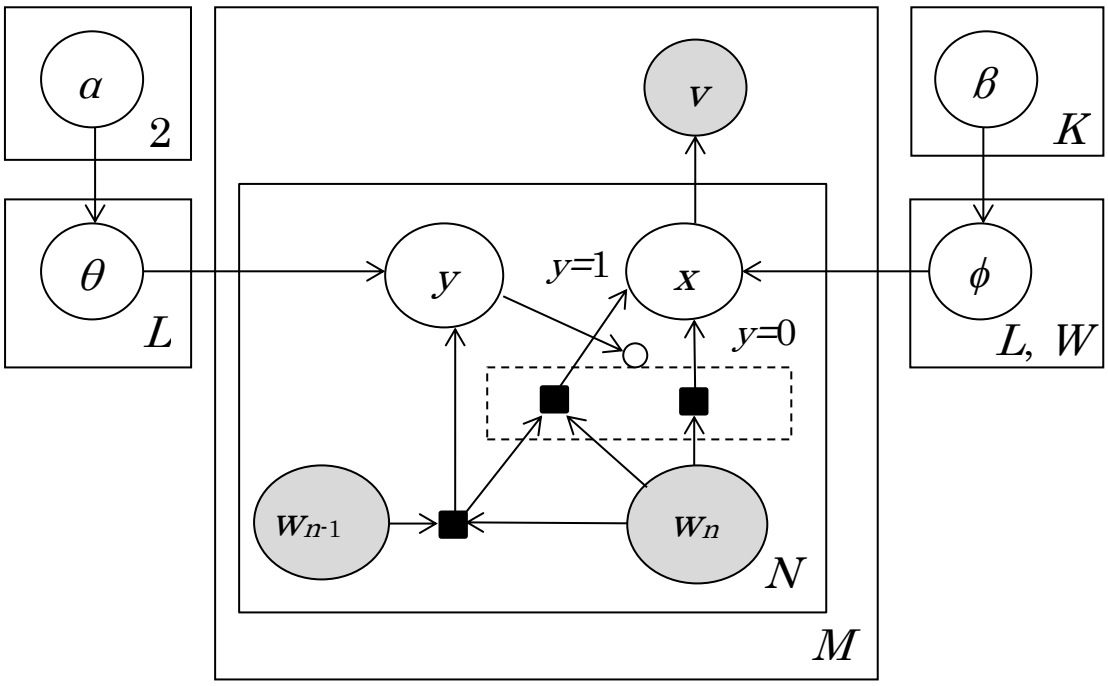

Figure 3-2. Graphical representation of *LVM*.

We adopt a Bayesian approach to embed these properties in our model. The prior distribution of $(\theta_0^{(a,b)}, \theta_1^{(a,b)})$ is 2-dimensional Dirichlet distribution (beta distribution) $Dir(\alpha_0, \alpha_1)$. To reflect the property (b) above, we set the meta-parameters α_0 and α_1 as follows: $0 < \alpha_0, \alpha_1 < 1$ and $\alpha_0 > \alpha_1$. The prior distributions of $(\phi_0^{(b)}, \phi_1^{(b)}, \cdots, \phi_{21}^{(b)})$ and $(\phi_0^{(a,b)}, \phi_1^{(a,b)}, \cdots, \phi_{21}^{(a,b)})$ are 22-dimensional Dirichlet distributions $Dir(\beta_0, \beta_1, \ldots, \beta_{21})$. To reflect the property (a) above, we set the meta-parameters as follows:

$$0 < \beta_0, \beta_1, \cdots, \beta_{21} < 1, \text{ and } \beta_0 > \beta_1 + \beta_2 + \cdots + \beta_{21}.$$

Furthermore, to keep the number of meta-parameters small, we set the following restrictions:

$$\begin{aligned} &\alpha_0 = \alpha, \ \ \alpha_1 = \gamma\alpha, \ \beta_0 = \alpha, \ \ \beta_i = \gamma\alpha / 21 \\ &(i = 1, 2, \cdots, 21, \ 0 < \alpha < 1 \text{ and } 0 < \gamma < 1). \end{aligned} \tag{3}$$

Thus, the free meta-parameters are only α and γ.

When the word-level values $\mathbf{x}$ is determined, the sentence-level values v is uniquely determined as $v = x_1 \oplus x_2 \oplus \cdots \oplus x_N$. Therefore, the probability of $(\mathbf{x}, \mathbf{y}, v)$ given $\mathbf{w}$ is then:

$$P(\mathbf{x}, \mathbf{y}, v \mid \mathbf{w}, \theta, \phi) = \begin{cases} P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \theta, \phi) & ; \ \mathbf{x} \in \chi^N(v) \\ 0 & ; \ otherwise \end{cases}.$$

The probability of $(\mathbf{x}, \mathbf{y})$ given $(\mathbf{w}, v)$ is therefore:

$$P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, v, \theta, \phi) \propto \begin{cases} P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \theta, \phi) & ; \ \mathbf{x} \in \chi^N(v) \\ 0 & ; \ otherwise \end{cases}. \tag{4}$$

(2) Estimation of Values

Let $(\mathbf{W}, V)$ be a collection of sentences and their values. $\mathbf{W} = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(M)})$, where $\mathbf{w}^{(m)}$ is a sentence, and $V = (v_1, v_2, \ldots v_M)$, where v_m is the value(s) of the m-th sentence. The n-th word of $\mathbf{w}^{(m)}$ is denoted $w_n^{(m)}$, and the length of m-th sentence is denoted N_m. The collection $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(M)})$ is denoted $\mathbf{X}$, and the collection $(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(M)})$ is denoted $\mathbf{Y}$ in a like manner. We can get the probability of $(\mathbf{X}, \mathbf{Y})$ given $(\mathbf{W}, V)$ from (4) as follows:

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y} \mid \mathbf{W}, V, \theta, \phi) &= \prod_{m=1}^{M} P(\mathbf{x}^{(m)}, \mathbf{y}^{(m)} \mid \mathbf{w}^{(m)}, v_m, \theta, \phi) \\ &\propto \begin{cases} \prod_{a,b} \prod_{u \in \{0,1\}} \{\theta_u^{(a,b)}\}^{C_Y((a,b),u)} \times \prod_{b} \prod_{t} \{\phi_t^{(b)}\}^{C_X(b,t,0)} \\ \quad \times \prod_{a,b} \prod_{t} \{\phi_t^{(a,b)}\}^{C_X((a,b),t,1)} & ; \ \mathbf{x}^{(m)} \in \chi^{N_m}(v_m) \text{ for all } m, \\ \qquad 0 & ; \ otherwise \end{cases} \end{aligned}$$

where $C_Y((a, b), u)$ is the number of times u has been assigned to a two-word sequence (a, b) as the value of context indicator y, $C_X(b, t, 0)$ is the number of times value μ_t has been assigned to word b without the influence of the previous word, and $C_X((a, b), t, 1)$ is the number of times value μ_t has been assigned to the word b with the influence of the previous word a.

When $\mathbf{x}^{(m)} \in \chi^{N_m}(v_m)$ for all m=1, 2, ..., and M, we get the following formula by calculating the marginal probability:

$$
\begin{aligned}
&P(\mathbf{X}, \mathbf{Y} \mid \mathbf{W}, V, \alpha, \gamma) \\
&= \int P(\mathbf{X}, \mathbf{Y} \mid \mathbf{W}, V, \theta, \phi)\pi(\theta \mid \alpha, \gamma)\pi(\phi \mid \alpha, \gamma)\, d\theta\, d\phi \\
&\propto \prod_{a,b} \frac{\Gamma(\sum_{u\in\{0,1\}} \alpha_u)}{\prod_{u\in\{0,1\}} \Gamma(\alpha_u)} \frac{\prod_{u\in\{0,1\}} \Gamma(C_Y((a,b),u)+\alpha_u)}{\Gamma(\sum_{u\in\{0,1\}} \{C_Y((a,b),u)+\alpha_u\})} \\
&\times \prod_{b} \frac{\Gamma(\sum_t \beta_t)}{\prod_t \Gamma(\beta_t)} \frac{\prod_t \Gamma(C_X(b,t,0)+\beta_t)}{\Gamma(\sum_t \{C_X(b,t,0)+\beta_t\})} \\
&\times \prod_{a,b} \frac{\Gamma(\sum_t \beta_t)}{\prod_t \Gamma(\beta_t)} \frac{\prod_t \Gamma(C_X((a,b),t,1)+\beta_t)}{\Gamma(\sum_t \{C_X((a,b),t,1)+\beta_t\})}\ ,
\end{aligned}
\tag{5}
$$

where $\pi(\theta \mid \alpha, \gamma)$ and $\pi(\phi \mid \alpha, \gamma)$ are the prior distribution of θ and the prior distribution of ϕ, respectively, and $\Gamma(\cdot)$ is the gamma function.

We can estimate the value(s) for a sentence $\mathbf{w}$ that has N words:

$$
\begin{aligned}
values(\mathbf{w}) &= \arg\max_{v} P(v \mid \mathbf{w}, \alpha, \gamma) \\
&= \arg\max_{v} \sum_{\mathbf{y}\in\{0,1\}^N} \sum_{\mathbf{x}\in\chi^N(v)} P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \alpha, \gamma).
\end{aligned}
$$

Also we can estimate that $\mathbf{w}$ has the j-th value of the six, when

$$
\sum_{v:(v)_j=1} \sum_{\mathbf{y}\in\{0,1\}^N} \sum_{\mathbf{x}\in\chi^N(v)} P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \alpha, \gamma) \geq \frac{1}{2}.
\tag{6}
$$

That is, whether a sentence has the j-th value is determined for each j separately. This means that the comparison among SVM, sLDA and *LVM* is fair enough because both judgments do not take into account combination of values.

(3) Posterior probabilities by Gibbs sampling

We need the probabilities $P(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \alpha, \gamma)$ for every $\mathbf{x}$ and $\mathbf{y}$ to estimate the values of a sentence $\mathbf{w}$. These are the predictive posterior probabilities after the training data $(\mathbf{W}, V)$, $\hat{P}(\mathbf{x},\mathbf{y} \mid \mathbf{w},\mathbf{W},V,\alpha,\gamma)$, to be exact. The predictive posterior probabilities are calculated by integrating out θ and ϕ as follows:

$$
\begin{aligned}
&\hat{P}(\mathbf{x},\mathbf{y} \mid \mathbf{w},\mathbf{W},V,\alpha,\gamma) \\
&= \int P(\mathbf{x},\mathbf{y} \mid \mathbf{w},\theta,\phi) \cdot \pi(\theta,\phi \mid \mathbf{W},V,\alpha,\gamma)\, d\theta\, d\phi \\
&= \sum_{\mathbf{X}} \sum_{\mathbf{Y}} \frac{P(\mathbf{X},\mathbf{Y},V \mid \mathbf{W},\alpha,\gamma)}{P(V \mid \mathbf{W},\alpha,\gamma)} \int P(\mathbf{x},\mathbf{y} \mid \mathbf{w},\theta,\phi) \frac{P(\mathbf{X},\mathbf{Y},V,\theta,\phi \mid \mathbf{W},\alpha,\gamma)}{P(\mathbf{X},\mathbf{Y},V \mid \mathbf{W},\alpha,\gamma)}\, d\theta\, d\phi\,,
\end{aligned}
$$

where $\pi(\theta,\phi \mid \mathbf{W},V,\alpha,\gamma)$ is the posterior probability after giving the training data $(\mathbf{W},V)$.

In the last formula above,

$$
\int P(\mathbf{x},\mathbf{y} \mid \mathbf{w},\theta,\phi) \frac{P(\mathbf{X},\mathbf{Y},V,\theta,\phi \mid \mathbf{W},\alpha,\gamma)}{P(\mathbf{X},\mathbf{Y},V \mid \mathbf{W},\alpha,\gamma)}\, d\theta\, d\phi
$$

is the predictive posterior probability $\hat{P}(\mathbf{x},\mathbf{y} \mid \mathbf{w},\mathbf{W},V,\mathbf{X},\mathbf{Y},\alpha,\gamma)$, after observation $(\mathbf{W},V,\mathbf{X},\mathbf{Y})$.

For instance, when $\mathbf{w} = (a, b)$ $(a \neq b)$, $\mathbf{x} = (\mu_j, \mu_k)$, and $\mathbf{y} = (0, 1)$, $\hat{P}(\mathbf{x},\mathbf{y} \mid \mathbf{w},\mathbf{W},V,\mathbf{X},\mathbf{Y},\alpha,\gamma)$ can be calculated by integrating out θ and ϕ as follows:

$$
\begin{aligned}
&\int \theta_0^{(\$,a)} \theta_1^{(a,b)} \phi_j^{(a)} \phi_k^{(a,b)} \frac{P(\mathbf{X},\mathbf{Y},V,\theta,\phi \mid \mathbf{W},\alpha,\gamma)}{P(\mathbf{X},\mathbf{Y},V \mid \mathbf{W},\alpha,\gamma)}\, d\theta\, d\phi \\
&= 1 \times \frac{C_Y((a,b),1)+\alpha_1}{\sum_{u\in\{0,1\}}\{C_Y((a,b),u)+\alpha_u\}} \times \frac{C_X(a,j,0)+\beta_j}{\sum_t\{C_X(a,t,0)+\beta_t\}} \times \frac{C_X((a,b),k,1)+\beta_k}{\sum_t\{C_X((a,b),t,1)+\beta_t\}}
\end{aligned}
$$

Also, when $\mathbf{w} = (a, a)$, $\mathbf{x} = (\mu_j, \mu_k)$ $(\mu_j \neq \mu_k)$, and $\mathbf{y} = (0, 0)$, $\hat{P}(\mathbf{x},\mathbf{y} \mid \mathbf{w},\mathbf{W},V,\mathbf{X},\mathbf{Y},\alpha,\gamma)$ can be calculated:

$$
\begin{aligned}
&\int \theta_0^{(\$,a)} \theta_0^{(a,a)} \phi_j^{(a)} \phi_k^{(a)} \frac{P(\mathbf{X},\mathbf{Y},V,\theta,\phi \mid \mathbf{W},\alpha,\gamma)}{P(\mathbf{X},\mathbf{Y},V \mid \mathbf{W},\alpha,\gamma)}\, d\theta\, d\phi \\
&= 1 \times \frac{C_Y((a,a),0)+\alpha_0}{\sum_{u\in\{0,1\}}\{C_Y((a,a),u)+\alpha_u\}} \times \frac{C_X(a,j,0)+\beta_j}{\sum_t\{C_X(a,t,0)+\beta_t\}} \times \frac{C_X(a,k,0)+\beta_k}{\sum_t\{C_X(a,t,0)+\beta_t\}}.
\end{aligned}
$$

However, in the case where $\mathbf{w} = (a, a)$, $\mathbf{x} = (\mu_j, \mu_j)$, and $\mathbf{y} = (0, 0)$, $\hat{P}(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \mathbf{W}, V, \mathbf{X}, \mathbf{Y}, \alpha, \gamma)$ becomes:

$$\int \theta_0^{(\$,a)} \theta_0^{(a,a)} \phi_j^{(a)} \phi_j^{(a)} \frac{P(\mathbf{X}, \mathbf{Y}, V, \theta, \phi \mid \mathbf{W}, \alpha, \gamma)}{P(\mathbf{X}, \mathbf{Y}, V \mid \mathbf{W}, \alpha, \gamma)} d\theta\, d\phi$$
$$= 1 \times \frac{C_Y((a,a),0) + \alpha_0}{\sum_{u \in \{0,1\}} \{C_Y((a,a),u) + \alpha_u\}} \times \frac{C_X(a,j,0) + \beta_j + 1}{\sum_t \{C_X(a,t,0) + \beta_t\} + 1} \times \frac{C_X(a,j,0) + \beta_j}{\sum_t \{C_X(a,t,0) + \beta_t\}},$$

because of the property of the Γ function: Γ(z+2) = (z+1)zΓ(z). When there are more than two occurrences for one unique word, we have to take into account a large number of combinations for the theoretically-derived calculation. Then we approximate above calculation as:

$$\int \{\phi_j^{(a)}\}^2 \frac{P(\mathbf{X}, \mathbf{Y}, V, \theta, \phi \mid \mathbf{W}, \alpha, \gamma)}{P(\mathbf{X}, \mathbf{Y}, V \mid \mathbf{W}, \alpha, \gamma)} d\theta\, d\phi \cong \left\{ \int \phi_j^{(a)} \frac{P(\mathbf{X}, \mathbf{Y}, V, \theta, \phi \mid \mathbf{W}, \alpha, \gamma)}{P(\mathbf{X}, \mathbf{Y}, V \mid \mathbf{W}, \alpha, \gamma)} d\theta\, d\phi \right\}^2 .$$

By this approximation, the predictive probability in the last case above becomes as follows:

$$\int \theta_0^{(\$,a)} \theta_0^{(a,a)} \phi_j^{(a)} \phi_j^{(a)} \frac{P(\mathbf{X}, \mathbf{Y}, V, \theta, \phi \mid \mathbf{W}, \alpha, \gamma)}{P(\mathbf{X}, \mathbf{Y}, V \mid \mathbf{W}, \alpha, \gamma)} d\theta\, d\phi$$
$$= 1 \times \frac{C_Y((a,a),0) + \alpha_0}{\sum_{u \in \{0,1\}} \{C_Y((a,a),u) + \alpha_u\}} \times \frac{C_X(a,j,0) + \beta_j}{\sum_t \{C_X(a,t,0) + \beta_t\}} \times \frac{C_X(a,k,0) + \beta_j}{\sum_t \{C_X(a,t,0) + \beta_t\}}.$$

We found the difference between the theoretically-derived and the approximate calculation was not statistically significant in preliminary experiments. We therefore used the approximate calculation in our actual implementation for efficiency reasons.

Therefore, the predictive posterior probabilities become:

$$\hat{P}(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \mathbf{W}, V, \alpha, \gamma)$$
$$= \sum_{\mathbf{Y}} \sum_{\mathbf{X}} P(\mathbf{X}, \mathbf{Y} \mid \mathbf{W}, V, \alpha, \gamma) \cdot \hat{P}(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \mathbf{W}, V, \mathbf{X}, \mathbf{Y}, \alpha, \gamma). \tag{7}$$

By the law of large numbers, equation (7) can be approximated as follows:

$$\frac{1}{T} \sum_{t=1}^{T} \hat{P}(\mathbf{x}, \mathbf{y} \mid \mathbf{w}, \mathbf{W}, V, \mathbf{X}(t_0 + t), \mathbf{Y}(t_0 + t), \alpha, \gamma) \tag{8}$$

In equation (8), $(\mathbf{X}(s), \mathbf{Y}(s))$ is the s-th sample of $(\mathbf{X}, \mathbf{Y})$ that is drawn according to the posterior probability $P(\mathbf{X}, \mathbf{Y} \mid \mathbf{W}, V, \alpha, \gamma)$ given by a Gibbs sampler. We get the conditional probability used in Gibbs sampling from equation (5) as follows:

When $(x_1^{(m)} \oplus \cdots \oplus x_{n-1}^{(m)} \oplus \mu_j \oplus x_{n+1}^{(m)} \oplus \cdots \oplus x_{N_m}^{(m)}) \neq v_m$:

$$P(x_n^{(m)} = \mu_j, y_n^{(m)} = u \mid \mathbf{X}_{-(m,n)}, \mathbf{Y}_{-(m,n)}, \mathbf{W}, V, \alpha, \gamma) = 0$$

When $(x_1^{(m)} \oplus \cdots \oplus x_{n-1}^{(m)} \oplus \mu_j \oplus x_{n+1}^{(m)} \oplus \cdots \oplus x_{N_m}^{(m)}) = v_m$:

$$\begin{aligned} &P(x_n^{(m)} = \mu_j, y_n^{(m)} = 0 \mid \mathbf{X}_{-(m,n)}, \mathbf{Y}_{-(m,n)}, \mathbf{W}, V, \alpha, \gamma) \\ &= \frac{C_Y^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}),0) + \alpha_0}{\sum_{u \in \{0,1\}} \{C_Y^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}),u) + \alpha_u\}} \\ &\quad \times \frac{C_X^{-(m,n)}(w_n^{(m)}, j, 0) + \beta_j}{\sum_t \{C_X^{-(m,n)}(w_n^{(m)}, t, 0) + \beta_t\}}, \end{aligned}$$

$$\begin{aligned} &P(x_n^{(m)} = \mu_j, y_n^{(m)} = 1 \mid \mathbf{X}_{-(m,n)}, \mathbf{Y}_{-(m,n)}, \mathbf{W}, V, \alpha, \gamma) \\ &= \frac{C_Y^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}),1) + \alpha_1}{\sum_{u \in \{0,1\}} \{C_Y^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}),u) + \alpha_u\}} \\ &\quad \times \frac{C_X^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}), j, 1) + \beta_j}{\sum_t \{C_X^{-(m,n)}((w_{n-1}^{(m)}, w_n^{(m)}), t, 1) + \beta_t\}}, \end{aligned}$$

where $\mathbf{X}_{-(m,n)}$ is $\mathbf{X}$ from which $x_n^{(m)}$ is removed, and $C_X^{-(m,n)}(\cdot)$ is a count that does not include the current assignment of $x_n^{(m)}$. The same holds for $\mathbf{Y}_{-(m,n)}$ and $C_Y^{-(m,n)}(\cdot)$ with $y_n^{(m)}$.

3.6 Experiments

In this section, we describe our experiment design, report classifier effectiveness, and compare our automated results to those of a human annotator.

3.6.1 Experiment Design

It is important to examine whether the classifier works actually well in real-world setting through text structures. In our case, the corpus is provided by documents and annotated sentences are elements of a document. In our human annotation task is done for each document, even in the case where the annotation is attached for each sentence. In order to reflect this real situation, we use the sentences in a document unit for both training set and test set, which are divided at document boundaries.

Thus we use 102-fold document-scale cross-validation (except in Table 3-2, where in preliminary experiments we had not grouped sentences by document). 102-fold cross-validation seeks to model the case in which some set of 101 documents have been annotated as training data and we are interested in the degree to which the machine can automatically code all future documents. To select the meta-parameters for each fold, we use average one document size of sentences as held-out for development testing and rest

of approximately 100 document size of sentences for development training. We perform a parameter sweep by training on all sentences in the development training set and then testing on all sentences in the one development testing document to select the meta-parameters α and γ that yield the best F_1, sweeping both parameters across 0.05, 0.1, 0.2, 0.5 and 0.9. The 101-document training set is trained using the best α and γ, and the resulting model is used to classify the sentences in the test set.

For Gibbs sampling we used 50,000 trials. Thirty percent of those trials were treated as the burn-in period. We used 1-for-3 samples of them as $(\mathbf{X}(t_0+1), \mathbf{Y}(t_0+1))$, $(\mathbf{X}(t_0+2), \mathbf{Y}(t_0+2))$, and so on to calculate equation (8). These parameters were empirically determined in preliminary experiments on development data. We apply the same process to determine the frequency threshold η of bigram features (use if frequency ≥ η, a meta-parameter for SVM[6]) and to determine the meta-parameters for sLDA[7].

In order to examine influence of the previous word, we compare our *LVM* with *LVM* ($y_n=0$) which is our model without any influence from the previous word (i.e., with the context indicator y_n in the equation (1) always zero. The meta-parameters were same as for *LVM*.) We also compare our models with two types of SVM as fair baselines, SVM(w) and SVM(w, b). SVM(w) uses only word features, and SVM(w, b) uses word and bigram features. We use 2nd-degree polynomial kernel for SVM(w) and linear kernel for SVM(w, b), that kernels are determined respectively in experiments.

sLDA (Blei and Mcauliffe 2007) is a general supervised method but it inherited the property of LDA (Blei et al. 2003; Griffiths et al. 2004) which is a generative model for "documents" so that multiple topics are responsible for the words occurring in a single document. When we apply sLDA our test corpus, we assume that one sentence is regarded as a document. This setting might lose reliability of sLDA's behavior, because the expected number of words which have values in a sentence is a few. However, sLDA is a representative supervised probabilistic model, so we investigate how it works in the actual experiment.

3.6.2 Results

Table 3-2 shows results for SVM(w), SVM(w, b), sLDA, *LVM* ($y_n=0$) and *LVM*. *LVM* is much better than sLDA. We can see that even *LVM* ($y_n=0$) outperforms SVM(b) and SVM(w, b) (significantly for SVM(b), but not significantly for SVM(w, b)).

Table 3-3 shows classifier effectiveness by 102-fold document cross-validation. We omit *honor* from these micro-averaged results because no classifier did well for that category due to a scarcity of annotations for that value in our corpus, as illustrated in Table 2-2. As can be seen, *LVM* apparently outperforms SVM(w, b). This is also true for sLDA, even when the number of topics is set to 22, which is the closest approximation to our model.

---

[6] http://chasen.org/~taku/software/TinySVM/

[7] http://www.cs.cmu.edu/~chongw/slda/

In sLDA, the response is regressed on the topic proportions, while the SVM calculates the weights for the response directly from words. We believe the reason why sLDA works so badly is as follows: (1) it is a model for "document" but not for "sentence" as we mention in the section 3.6.1; (2) linear regression of the latent variables for words to explain the response is not as well suited to our very sparse data as our estimation of the sentence-level values by a bitwise OR of the word-level values is.

Table 3-4 shows per-category effectiveness measures for the SVM and for our *LVM*, respectively. For each comparison across the two classifiers, the bolded value is the higher of the two results. This is always true for F_1, except for the case of the category with the fewest training examples, *honor*. As for the comparison between SVM(w) and *LVM*, p-value 3.94×10^{-9} of the difference in the average F_1 between them, which is calculated by an exact randomization test (Noreen 1989; Smucker 2007; Japkowicz and Shah 2011; Scheible 2014), suggests that the equality can be rejected in significance level 0.01.

As Table 3-3 shows, SVM(w) and SVM(w, b) achieve nearly identical F_1 with 102-fold document cross-validation (the same condition reported in Table 3-4, which models the actual annotation process), with SVM(w) yielding F_1=0.7166 and SVM(w, b) yielding 0.7154. We therefore chose SVM(w) with the numerically higher score as the illustrative baseline for Table 3-4.

The value *honor* is omitted from the averages in Tables 3-2 and 3-3 because we focus our analysis of those tables on relative comparisons between usable classifiers. As Table 3-4 shows, the recall for *honor* is too low (0.28 in SVM and 0.16 in *LVM*) for practical application.

To better understand the behavior of *LVM* on this collection, we have looked into the estimated word-level values as the first step of qualitative analysis. The social scientists collaborating on this research identified cue words used to invoke particular values during the annotation process. For example, "American consumers will lose basic Internet *freedoms*, the engine of *innovation* will be hobbled, and our global competitiveness will be compromised" which is annotated with *freedom*, *innovation*, and *wealth* as sentence-level values. The values names serve as good cue words, and *LVM* assigned the appropriate values for the words "freedom" and "innovation". As for *wealth*, *LVM* estimated that "competitiveness" has the word-level value *wealth* with influence from the previous word "global". We assumed that each word in a sentence has at-most-two values, and *LVM* aggregates the word-level values above then correctly estimated all three sentence-level values for the sentence. We plan to conduct more detailed qualitative analysis in our future work.

Note that micro-averages of *LVM* and *LVM* (y_n=0) described in Tables 3-2, 3-3, and 3-4 are corrected from those published in our CIKM 2014 paper (Takayama et al. 2004), especially for *honor* in Table 3-4. A statistical testing method using this section and the next section is also corrected from a z-test to an exact randomization test.

Table 3-2. Classifier effectiveness (micro-averaged without *honor*, 3×10-fold sentence cross-validation).

Method	Precision	Recall	F_1
SVM(w)	0.7924	0.6802	0.7320
SVM(w, b)	0.7784	0.6988	0.7365
sLDA	0.7016	0.4821	0.5715
LVM ($y_n = 0$)	0.7916	0.6931	0.7391
LVM	**0.8001**	**0.7133**	**0.7542**

The meta-parameters for sLDA: α= 0.05, 0.1, 0.2, 0.5 or 0.9 (fixed at initial α), the number of topics K=16, 22, 32, 64, 96 or 128. The meta-parameters for SVM(w, b): Bigram frequency threshold η = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ∞(= w/o bigrams).

Table 3-3. Classifier effectiveness (micro-averaged without *honor*, 102-document cross-validation).

Method	Precision	Recall	F_1
SVM(w)	0.7784	0.6638	0.7166
SVM(w, b)	0.7535	0.6809	0.7154
sLDA	0.6875	0.4591	0.5506
LVM ($y_n = 0$)	**0.7910**	0.6785	0.7305
LVM	0.7866	**0.6902**	**0.7353**

The meta-parameters for classifiers are the same as those in Table 3-2.

Table 3-4. Per-category effectiveness (102-document cross-validation, micro-averaged).

Value	Precision		Recall		F_1	
	SVM(w)	*LVM*	SVM(w)	*LVM*	SVM(w)	*LVM*
wealth	0.7859	**0.7908**	0.6977	**0.7402**	0.7392	**0.7646**
social order	**0.8235**	0.7803	0.7587	**0.8174**	0.7898	**0.7984**
justice	0.7275	**0.7823**	**0.5558**	0.5483	0.6302	**0.6447**
freedom	0.7461	**0.7911**	0.6654	**0.6729**	0.7035	**0.7272**
innovation	**0.8139**	0.7898	0.5629	**0.5756**	0.6655	**0.6659**
honor	0.4324	**0.5085**	**0.2019**	0.0946	**0.2753**	0.1596
average	0.7730	**0.7849**	0.6510	**0.6737**	0.7068	**0.7251**

3.6.3 Comparison with Human Annotation

Because human values are unobservable private states rather than observable facts (Weibe 1994), we see the annotator's task as rendering an opinion about which values a statement reflects, and the system's task as replicating that result. As our inter-annotator agreement in Table 2-2 indicates, well trained and well qualified people will sometimes make different judgments about the same sentence. To see how our *LVM* compares with human annotator on a per-category basis, we ran experiments with the 20 documents (2,430 sentences) annotated by a second annotator with sufficient agreement with the first annotator as described in Section 2.2.

For this experiment, we trained *LVM* on the remaining 82 documents with meta-parameters: $\alpha = 0.2$, $\gamma = 0.9$ (most frequently selected meta-parameters during document cross-validation). For comparability, we treat the first annotator's annotations of those 20 documents as correct, and we compute effectiveness as if the second annotator were a classifier. The results are shown in Table 3-5.

Table 3-5. Human "classifier" and *LVM* effectiveness (same 20 test documents., micro-averaged).

Value	Precision		Recall		F_1	
	Human	*LVM*	Human	*LVM*	Human	*LVM*
wealth	0.735	0.816	0.871	0.681	0.797	0.743
social order	0.775	0.748	0.759	0.820	0.767	0.782
justice	0.664	0.739	0.464	0.544	0.546	0.627
freedom	0.681	0.780	0.768	0.704	0.722	0.740
innovation	0.764	0.736	0.720	0.640	0.741	0.685
honor	0.395	0.571	0.553	0.094	0.461	0.162
average	0.712	0.772	0.732	0.668	0.722	0.716

Although human performance is not necessarily an upper bound on performance (because the classifier has more access to evidence about how one annotator makes decisions than another human would), we see it as a useful reference because the utility of our classifier depends on its relative costs and benefits when compared to the alternative for coding at large scales, which would be to hire many annotators. Our results show that automation can achieve results similar to human annotation, but at a lower cost (in terms of human effort).

The difference in the average F_1 between human and *LVM*, p-value 0.418 which is calculated by an exact randomization test (Noreen 1989; Smucker 2007; Japkowicz and

Shah 2011; Scheible 2014), suggests that the equality cannot be rejected in significance level 0.01. This means that *LVM* effectiveness is statistically indistinguishable from the human classifier. As can be seen, *LVM* does about as well as our human second annotator on average, and it does substantially better in both precision and recall (and thus in F_1) than the second annotator on *justice*. Notably, *honor* is markedly less problematic for the human second annotator than for *LVM*.

3.7 Summary

We have proposed a word-level probabilistic latent variable model for detecting the sentence-level human values reflected in prepared statements on a contentious political issue. The model treats the words in a sentence as having been chosen based on specific human values, and the values reflected by each sentence thus can be estimated by aggregating the values associated with each word. We have achieved the highest reported sentence classification effectiveness $F_1 = 0.735$ (micro-averaged without *honor*) in 102-document cross-validation, which is a 3% relative improvement over SVM(w) that does not take account of sequential dependencies between words, as our model does. *LVM* also improved over SVM(w, b), which uses bigram features.

Our model can determine the human value(s) x_n for the word w_n in light of the influence of the previous word w_{n-1}. It is natural to next consider that word w_n's value(s) x_n might also be influenced by the both previous word w_{n-1} and following word w_{n+1}. This more complex model may suffer from sparsity, however. We might also explore using longer-distance syntactic dependencies found by a dependency parser, but since dependency parsing is imperfect, proximity features will likely continue to offer some benefit.

Chapter 4

Qualitative Analysis Using Values-dictionary

4.1 Introduction

We briefly look back the corpus used in Chapters 2 and 3 again. In prior work Cheng (2012) has described the process that they used to manually annotate 102 prepared testimonies related to net neutrality. They started with a meta-inventory of human values (MIHV) that had originally been developed and assembled by Cheng and Fleischmann (2010) from many pre-existing values inventories and they iteratively tailored a domain-specific values inventory through four rounds of coding. This process resulted in a set of six value categories relevant to the net neutrality debate that human annotators could reliably annotate: *freedom*, *honor*, *innovation*, *justice*, *social order*, and *wealth* (Cheng 2012).

Specifically, all 102 prepared testimonies were annotated by a social scientist for values at the sentence level. Coding involved detecting explicit and implicit invocations of values, and included statements that expressed positive, negative, and neutral sentiment toward those testimonies. A sentence could reflect multiple values, or none. Twenty of these testimonies were coded independently by two annotators. Substantial agreement was achieved for *freedom*, *innovation*, *social order*, and *wealth*, and moderate agreement was achieved for *honor* and *justice* described in Table 2-2 (Cheng 2012).

In previous chapters, we have demonstrated the ability to train classifiers for the assignment of values to sentences by using a suite of latent variable models and support vector machines, one for each value (Takayama et al. 2013, 2014). The resulting systems agree with human annotations nearly as often as another human annotator would for five of the six values (specifically, all values other than *honor*, which is the value least often annotated in our collection).

Latent variable models (Chapter 3) and support vector machines (Chapter 2 and 3) can be effective, but they reason very differently than people do and thus they are not well suited to the development of explanations for their results. For this chapter, we build up a "values-dictionary" by using a simplified model of *LVM*, that uses easily interpreted classification rules.

In the social psychology field, a dictionary called LIWC (Tausczik and Pennebaker 2010) provides a set of general terminology and a category database for the psychological meaning of words to analyze text with computers. What distinguishes between LIWC and our proposed values-dictionary is that we concentrate on human values and provide automatic dictionary construction method that depends on the targeting corpus to be analyzed.

4.2 Dictionary Extraction Method using Simulated Annealing

For each single word or each pair of words we learn an entry in a values-dictionary that simply tells us which values to assign. Then whenever we see that word or pair of adjacent words in a sentence, we assign the corresponding values to that sentence. To avoid guessing every value for every sentence, the values-dictionary must be sparse – it must assign no values to most words. Given enough time, we could try every possible values-dictionary (i.e., every combination of values for every word – for example, maybe when we see "financial" in a document we should guess the values *wealth* and *social order*?) and then pick the best one (by checking to see how well each values-dictionary would work on some subset of our annotated interviews that we use for "training").

In practice, we use a simple machine learning technique, simulated annealing (Kirkpatrick et al. 1983), to efficiently explore the space of reasonable possibilities. This technique starts from a reasonable guess (the values we see most often with each word in annotated training sentences) and then iteratively tries small variations in the entries of the values-dictionary until no further improvement can be found (e.g., maybe seeing "financial" in a document should only cause us to guess the value *wealth*).

The formulation for the proposed method is the same as the one for *LVM* which is described in Section 3.5. A sentence $\mathbf{w}$ is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, where w_n is the n-th word in the sequence. The sentence has *sentence-level values* v, where $v \in \{0,1\}^6$, and the words that occur in that sentence have value(s) associated with each word in the sentence. Note that we restrict x_n to be an element in χ so that a word has at-most two values, as described in Section 3.5.1. If the word w_n has a *word-level value(s)* x, the sentence $\mathbf{w}$ also has the values x. On the other hand, if no word w_n in a sentence $\mathbf{w}$ has value(s) x, then the sentence $\mathbf{w}$ does not have values x. That is, the sentence-level values v are the result of bitwise OR for all word-level values corresponding to the words occur in that sentence. Under the above formulation, we apply a simulated annealing method to assign the human values for the words as follows:

(1) Calculate the maximum possible number of values for the word w under the following constraints. (a) The word frequency is higher than or equals to a given meta-parameter. (b) The ratio of {the sentence frequency with each value, where the word occurs} and {the

sentence frequency where the word occurs} is higher than another given meta-parameter.

(2) Repeat the following procedure until the values assignment becomes stable. (a) Try the modification of value(s) assignment randomly with small probability. (b) Compare the classification effectiveness F_1 score between the old values assignment and the new values assignment. (c) If the F_1 score is improved for the new values assignment, change the values assignment for the word.

(3) Gradually set the possibility of the chance of the modification smaller.

After the values assignment becomes stable, the words whose values assignment is not empty are extracted as the entries of values-dictionary.

Once the entries for values-dictionary are extracted, estimation of sentence-level values for unseen sentences in new documents is quite simple. For each word in a sentence, if it matches an entry of the values-dictionary, the word is assigned the values which the entry has. After the word-level values assignment process is finished, the sentence-level value(s) for that sentence is calculated by bitwise OR over all word-level values. This sentence-level values estimation process is deterministic, therefore, it is readily traceable even by a human.

Because single words may not be sufficiently informative as described in chapter 3, we also learn to associate adjacent two-word pairs with values by the above procedure. Then when we see a two-word pair (e.g., financial market) we guess the values associated with that pair (e.g., *wealth*) rather than the values associated with each word in the pair individually. Because we can accumulate more evidence for training if we treat variants of the same word in the same way, we make only a single entry in our values-dictionary for each word stem (e.g., market is the stem of market, markets, and marketing) or pair of stems; this is equivalent to making identical entries for any words or word pairs that share the same stem(s).

4.3 Experiments

Tables 4-1 and 4-2 show the classification effectiveness the proposed method adopting simulated annealing (denoted as *SA*), compared with SVM(w) and *LVM*, both described in Tables 3-3 and 3-4. The classification effectiveness in F_1 score of *SA* using the values-dictionary is lower than that of *LVM*. However, rather remarkably, this simple approach yields results that are about as accurate as those achieved by an SVM. The proposed *SA* achieves $F_1 = 0.7114$ (0.7211 without *honor*), which is about the same as the $F_1 = 0.7068$ (0.7166 without *honor*) as described in Tables 4-1 and 4-2.

Table 4-1. Classifier effectiveness (102-fold document cross-validation, with *honor*).

Classifier	Precision	Recall	F_1
SVM(w)	0.7730	0.6510	0.7068
LVM	**0.7849**	**0.6737**	**0.7251**
SA	0.7052	0.7177	0.7114

Table 4-2. Classifier effectiveness (102-fold document cross-validation, without *honor*).

Classifier	Precision	Recall	F_1
SVM(w)	0.7784	0.6638	0.7166
LVM	**0.7866**	0.6902	**0.7353**
SA	0.7058	**0.7370**	0.7211

These results are averages across 8,660 sentences based on 102-fold document cross-validation (in which we use every sentence in some set of 101 documents for training and then test on the sentences in the 102nd document, repeating the process 102 times with different sets of training and test documents). Unlike SVM or *LVM*, however, our proposed process produces a values-dictionary that we can now analyze.

4.4 Qualitative Analysis for each human value

This subsection shows the actual estimation results and qualitative analysis for them for each human value using the values-dictionary.

Qualitative analysis was conducted for randomly four selected examples for each six human values. Four examples are listed in order of true positive, true negative, false positive, and false negative for the targeted value. These indicators are represented in the first line of examples as 11, 00, 01 and 10, respectively, after the target value name. The first line also shows the estimated results for each value by the first character of value name, that is, 'f' for *freedom*, 'h' for *honor*, 'i' for *innovation*, 'j' for *justice*, 's' for *social order*, and 'w' for *wealth* respectively, and a one digit number. The number '1' after

the first character of the value name means that the corresponding value is detected and the number '0' means that the value is not detected. The brown color of this mnemonic code (one character and one number) represents a false positive for that value, and the green color represents a false negative. The last seven digit number in the first line is the sentence id in our test corpus.

The example sentences are described in italic fonts. Bolded words in example sentences are entries in the values-dictionary. We put first character of value names after the bolded words assigned the values. The blue words and the symbol "#" means that the single word is an entry in the values-dictionary, and the red word pairs and the symbol "##" means that the two word pair is an entry in the values-dictionary. One dictionary entry may have more than one human value. The maximum number of values for one word is three in this experiment. Note that the actual estimation is done by word stems and word stem pairs, however, this section shows examples using complete words for readability.

(1) Wealth

Wealth11 (true positive for **wealth)** (f1 h0 i0 j1 s1 w1) 0002118

*For this reason, it may well have been sensible for **Congress**#s to grant to cable owners an almost **unlimited**#w range of **freedom**#f to structure production decisions as they want, and develop cable offerings and **prices**#w as the **market**#w will **bear**#j.*

This example is fine for the detecting *wealth* by "prices as the market." (The word "unlimited" is marked as *wealth* also, however, "unlimited" in this context suggests *freedom*, not *wealth*.)

Wealth00 (true negative for **wealth)** (f1 h0 i0 j0 s1 w0) 0006051

*This tracks the approaches recently **proposed**#s by Senator DeMint in S.2113 , and by the Progress & **Freedom**#f Foundation in our Digital Age Communications **Act**#s project.*

This example is fine for no detection of *wealth*.

Wealth01 (false positive for **wealth)** (f0 h0 i0 j1 s1 w1) 0006123

*The network must implement and **enforce#s fairness**#j or else we have a state of anarchy where the "wants" of the few **constrain**#j the majority of the capacity that was intended for all **paying**#w users.*

The word "pay(ing)" may sometimes relate to *wealth*, but it is not in this context.

Wealth10 (false negative for **wealth)** (f0 h0 i0 j1 s0 w0) 0002761

*Put simply, they argue that if Verizon **degraded**#j access to a site or created a **discriminatory**#j "**fast lane**##j" that consumers **disliked**#j -- they would lose customers to the other network operators in the area.*

This sentence is annotated as *no wealth*, however, the part "they would lose customers to the other network operators" is the reason for an annotator to code it with *wealth*. This is illustrates a limitation of the current dictionary based method.

(2) Social Order

Social Order 11 (true positive for **social order)** (f0 h0 i0 j0 s1 w0) 0009205

*Historically, if a network operator is under no **legal**#s obligation to interconnect its network, voluntary interconnection rarely occurs.*

This example is fine for detecting *social order* by "legal."

Social Order 00 (true negative for **social order** (f1 h0 i0 j0 s0 w0) 0006398

*Comcast's **interference**#f occurs during all hours of every day, a fact which does not jive with the idea that it is somehow to responding to rare moments of congestion.*

This example is fine for no detection of *social order*.

Social Order 01 (false positive for **social order)** (f0 h0 i0 j1 s1 w1) 0000988

*As the **FCC**#s noted, such **prioritized**#j sharing should "both help to defray the **costs**#w of build-out and ensure that the spectrum is used **efficiently**#w.":*

The word "FCC" is sometimes appropriate for annotating *social order* but not for this case, thus this false positive is understandable.

Social Order 10 (false negative for **social order)** (f0 h0 i0 j0 s0 w1) 004585

*This Committee must help ensure that the Internet's **marketplace**#w of ideas and commerce remains effective and vibrant.*

The phrase "this Committee must help ensure that" invokes *social order*.

(3) Justice

Justice11 (**true positive** for **justice**) (f1 h0 i0 j1 s1 w1) 0005549

This is precisely the judicial analysis that Congress#s precluded#f in the 1996 Act##s , and this holding has done violence to remedial#s antitrust enforcement###f##j##s andcompetitive#f gains#w in the telecommunications marketplace#w.

This example is fine for detecting *justice* by "antitrust enforcement."

Justice00 (**true negative** for **justice**) (f0 h0 i0 j0 s1 w1) 0005692

They would also lead to the creation of a new bureaucracy to apply such rules#s and add layers of additional costs#w for dealing with the regulations#s and bureaucracy.

This example is fine for no detection of *justice*.

Justice01 (**false positive** for **justice**) (f1 h0 i0 j1 s0 w0) 0006332

But what about Free Press and the other petitioners who claim that limiting P2P#j harms free speech##f ?

The phrase "P2P harms free speech" could imply *justice*, but "P2P" alone should have no value. This example could be useful in annotation guidelines.

Justice10 (**false negative** for **justice**) (f0 h0 i0 j0 s1 w1) 0007396

The FCC#s has largely abandoned any oversight#s of special access pricing#w or terms and conditions, leaving the Bells free to raise competitors' costs#w with impunity.

The entire sentence talks about the lack of oversight leading to unjust competition; that is the reason for annotating this sentence with *justice*. This example illustrates a limitation of our values-dictionary based method.

(4) Freedom

Freedom11 (**true positive** for **freedom**) (**f1** h0 i0 j1 s1 w0) 0007382

*Furthermore, the **antitrust laws**##s rely on vigorous private **enforcement**#s -- in partnership with federal and state **antitrust enforcement**##f##j##s agencies to ensure that the **antitrust laws**##s are fully implemented.*

This example is fine for detecting *freedom* by "antitrust enforcement."

Freedom00 (**true negative** for **freedom**) (**f0** h0 i0 j0 s0 w0) 0002406

Third, broadband service providers should not be able to create private networks that are to superior to the Internet access they offer consumers generally.

This example is fine for no detection of *freedom*.

Freedom01 (**false positive** for **freedom**) (f1 h0 i0 j0 s0 w0) 0003980

*My name is Paul Morris, and I am the Executive Director of the Utah Telecommunication **Open**#f Infrastructure Agency, which we call "UTOPIA" for short.*

The word "open" is usually regarded as suggesting *freedom*, but this sentence is a factual statement. Thus this sentence should have no values.

Freedom10 (**false negative** for **freedom**) (f0 h0 i0 j0 s0 w0) 0007112

As long as the user utilized the standardized protocols, he could expect to send and receive packets to anyone else on the network in a completely understandable, predictable manner.

The phrase "he could to expect to send and receive to packets to anyone else on the network" is why the annotator coded *freedom* for this sentence. This is another example which reveals a limitation of a word dictionary-based method.

(5) Innovation

Innovation11 (**true positive** for **innovation**) (f0 h0 **i1** j0 s0 w0) 0008697

*The fidelity of the audio and video is so fine-tuned , it is as if the teacher and the **student**#i are in the same room, to able to discuss details about playing technique and musical phrasing.*

The word "student" is not a reason of annotating *innovation*. This is annotated as *innovation* because of the word "technique." In this case, the sentence-level values detection happens to be correct.

Innovation00 (**true negative** for **innovation**) (f1 h0 i0 j1 **s1** w1) 0008627

*Second, no **blocking**#j of **lawful**#s Internet content or services, so that consumers can be free to **access lawful##f content**##f or services, and so **startup**#w and other Internet companies can be to free to reach Internet consumers.*

This example is fine for no detection of *innovation*.

Innovation01 (**false positive** for **innovation**) (f0 h0 **i1 j1** s0 w0) 0000788

*At the time that network owners established the basic architectures for the major **broadband technologies**##i in the late 1990s, the Internet was dominated applications such as web browsing and e-mail that **adhered**#j to a client-server architecture.*

This is a factual statement here, but sometimes the two word sequence "broadband technology" could be annotated with *innovation*. The word "adhered" is learned as *innovation* by mistake.

Innovation10 (**false negative** for **innovation**) (f0 h0 **i0** j0 s0 w1) 0003854

*To me, as a scientist , it comes down ultimately to questions of physics and **economics**#w.*

The word "scientist", which is not listed in our values-dictionary, must have misled the coder to annotate this sentence with *innovation*. However, it is fine as no *innovation*. This example suggests to us that our values-dictionary will be used as a tool for minimizing such cases of mis-annotation. A machine classifier using values-dictionary can be able to be used for proof-reading.

(6) Honor

Honor11 (**true positive** for **honor**) (f0 **h1** i1 j0 s0 w1) 0001647

*These deployments will catapult New **Orleans**#h and Philadelphia into a worldwide leadership position in **technology**#i and will enable officials to meet the needs of their residents as well as enhance the visitor, tourism and **business**#w climate of those great cities.*

The word "Orleans" was leaaned as *honor* by mistake. The phrase "worldwide leadership position in technology" represents *honor*. This sentence happens to be judged correctly.

Honor00 (**true negative** for **honor**) (f0 **h0** i0 j1 s0 w1) 0006934

*Although **AOL**#j and GoodMail, which share the **profits**#w from the joint **venture**#w, claim that their program is **nondiscriminatory**#j , the facts tell us otherwise.*

This example is fine for no detection of *honor*.

Honor01 (**false positive** for **honor**) (f1 **h1** i0 j0 s0 w1) 0001636

*This network will serve all the citizens of New **Orleans**#h and Philadelphia by providing a **competitive**#f alternative to current broadband and dial-up Internet services -- at retail **rates**#w at or below the common **price**#w of **premium**#w dial-up Internet access.*

The word "Orleans" occurs again in this sentence because of the small numbers of positive examples for *honor*. That is the reason why the word "Orleans" suffers from overfitting.

Honor10 (**false negative** for **honor**) (f0 **h0** i0 j0 s0 w0) 0006189

We are pleased to be here today to encourage this committee continue to help unleash the full promise of Internet voice communications .

The entire sentence suggests this sentence should have the value *honor*. This example also reveals a limitation of a word dictionary-based method.

4.5 Summary

Our goal in this study is to use the resulting values-dictionary to now explore what our system has learned about how words are used to reflect values. As a contribution to the study of social science, we used an approximate optimization method, a simplified version of *LVM*, a model in which the human values associated with words (or word pairs) are uniquely determined, employing simulated annealing to extract a values-dictionary. With the values-dictionary, we can provide actual rich annotation examples that would be suitable for use in annotation guidelines as described in Section 4.4. The qualitative analysis described in this chapter shows that a human annotator might make a mistake even if well-trained. Our simple classifier using the values-dictionary can be used as a tool for avoiding annotation errors.

Chapter 5

Conclusion

5.1 Summary

The problem we addressed in this thesis, is estimation of human values that is reflected in statements in the documents regarding net neutrality, which consists of 8,660 sentences in 102 documents. The detection of sentence-level human values using supervised machine learning is done by estimating which values are reflected in which words that are the constituents of those sentences.

We first found that introducing word categories and word semantic classes fails to work properly for estimating human values in the experiments using SVMs with augmented feature vectors that contain hypernyms, synonyms, and associative words. This revealed that class-based generalization of words could not deal with training data sparsity. The fact led us design of a method that assigns the values directly to words or words pairs.

Then we proposed a probabilistic latent semantic variable model called *LVM* in which the classifier incorporates a mechanism whether estimation of human vales for a word is influenced by its previous word or estimation is done by the word itself without influence from the previous word. The classifier using *LVM* achieved the highest classification effectiveness (approximately three percent relative improvement compared with one by an SVM) for sentence-level human values in F_1 score. Furthermore, we found that the effectiveness of our proposed classifier is comparable with one of "human classifier," which is the annotation result by the second human annotator. This means that our proposed method can potentially substitute for human annotators.

Moreover, we built up a values-dictionary which consists of words and word pairs with values, by an approximate optimized method using simulated annealing. The classification effectiveness using the values-dictionary is lower than that of *LVM*, however, comparable with that of SVMs. More importantly, the entries in the values-dictionary serve as interpretable indicators of values, while the estimation results by *LVM* or SVMs is difficult to explain for social scientists. By using our values-dictionary, they can judge the values of the sentence deterministically depends on only the occurrences of the dictionary entries. In addition, we can provide rich annotation

examples for the annotation guideline by using the values-dictionary and a tool for preventing annotation errors.

5.2 Future Directions

This thesis discussed how we could implement estimation of human values using supervised machine learning. Finally, we describe several points as future directions of this study.

(1) Active learning using pseudo-negative examples

As a remaining issue of this thesis, we need to consider how we increase training data effectively for human values estimation, when classification effectiveness, by a classifier trained on the prepared data, is not sufficient. For this issue, we would like to apply our proposed active learning method for WSD, which uses "pseudo-negative examples" with high negative confidence scores, as described in Appendix B. This active learning method suggests an appropriate example to add into training data with human annotation for improving classification effectiveness.

(2) Extension to n-grams

We incorporated unigrams and bigrams into our proposed *LVM* in Chapter 3. For bigrams, we took only the influence of the previous word into account to determine the word-level values as local context. We should consider another probabilistic model which uses n-grams, instead of considering only unigrams and bigrams.

(3) Correlation among labels

Although we treated a multi-label problem in the thesis, however, we did not concern about correlation among labels (values) annotated for the same sentence. We know that there exist some kinds of correlations among values, thus we had better consider these correlations to seek further improvement in effectiveness of values estimation.

(4) Extension to estimating the ratio of human values in a document

When social scientists conduct content analysis, they want to obtain the ratio of human values that are reflected to a document. They analyze the relationship between the ratio of human values and the attributes assigned to the document for content analysis. The merits of *LVM* which we proposed in this thesis are: (1) the sentence-level human values are captured by an aggregation of word-level values; and (2) construct a model for estimating values by a word under the influence of the previous word. We can extend *LVM* to a generative model which has the parameters for assigning the ratio of values to a document. This generative model must have a high effectiveness for estimating the ratio of values that are assigned to a document, because the model inherits the merits of *LVM*.

(5) Constraint optimization methods

We used the traditional simulated annealing method, a kind of combinatorial optimization technique, in Chapter 4. This simple method achieved the same level of effectiveness compared with an SVM and provided useful information as a values-dictionary. Social scientists can use the dictionary entries for thematic analysis by assigning other labels and grouping the words. Therefore, we will investigate other optimizing methods such as integer programming and SAT for building up a better values-dictionary that facilitates content analysis.

(6) Other data sources and other domains

In this thesis, we used one data source: net neutrality corpus. To demonstrate the generality of our proposed methods, we should apply the methods to other corpora. One promising data source is the Internet news (net news) papers. The snippets of the net news papers consist of 30 words or less and similar contents are difficult to obtain because the news themes by their very nature have a short time span. The sparseness of net news data is similar to the net neutrality corpus. As another data source, traditional newspapers reflect a myriad of opinions about numerous social issues. When we demonstrate the effectiveness of the whole process of content analysis using human values, digitized traditional printed newspaper articles, which have certain related focused themes or domains, are prospective data sources.

Appendix

Appendix A

Definition and Annotation Scheme of Values

Table A shows the way we defined each annotated human value (Cheng 2012).

Table A. Definition and Annotation Scheme of Human Values.

Value	Definition
freedom	The condition of being free of restraints and encouraging competition; allowing individuals to have their own beliefs and to make their own choices; freedom from interference or influence of another or others; the quality of being autonomous and independent.
honor	Understanding of who you are and how you are perceived by others; a feeling of pride in oneself or one's organization, group, or nation and belief in one's own worth; accomplishment that is honored, esteemed, respected or well regarded by yourself or others.
innovation	The capacity to create or discover new things and new ideas that contribute to the advancement of knowledge and/or technology.
justice	The state of being treated equally and fairly, especially having the same rights, status, and opportunities; the process of settling a matter properly and fairly for all parties according to their capabilities and needs, especially protecting the weak and correcting any injustice; need for equal or fair distribution of resources, information, benefits, burdens, and power among the members of a society.
social order	Using the power of the government, military and/or legal system to protect the stability of society and/or to protect people from possible harms mentally or physically; acting in accordance with laws, regulations, and social norms.
wealth	An explicitly stated concern with or interest in pursuing economic goals such as money, material possessions, resources, and profit; focusing on the market value of a change, decision, or action; allocating resources appropriately and/or efficiently.

Appendix B

Active Learning with Pseudo Negative Examples for Word Sense Disambiguation

We studied the problem of building a WSD (word sense disambiguation) system using positive examples and unlabeled examples (Takayama et al. 2009). The motivation of this study is to extract a high accuracy dataset from a full text web search result that contains polysemous words for web mining applications. We assume a situation where a user of Internet search engines, would like to find only documents which contain a specific proper noun such as product name or person name in a specific sense (for example, "Tsubaki", a product name of shampoo). In this case, search target consist of both *positive examples* and the *negative examples*. Positive examples are documents that contain the word in the sense in mind ("Tsubaki" as a shampoo product name). Negative examples are documents that contain the word in different senses for which those are not intended ("Tsubaki" such as a flower name, clothes etc.) Fortunately, creation of positive examples in this situation is quite easy. We can retrieve positive examples from Web archive with high precision (but low recall) by manually augmenting queries with hypernyms or semantically related words (e.g., "Tsubaki AND shampoo" or "Tsubaki AND Shiseido"). In contrast, creation of negative examples is rather difficult because the user might not know how many senses for the word exist (e.g., people may not know "Tsubaki" has a sense of a horse race). For this reason, we would like to provide appropriate documents as training data to build up classifier effectively because we cannot read documents exhaustively. This is a binary-labeled classification problem.

We have proposed an active learning method with pseudo negative examples, as a solution of the above problem. The key feature of the proposed method is to estimate "pseudo negative" examples for obtaining reliable confidence score of unlabeled examples in order to select an appropriate example to add into training data with human annotation in active learning processes. The algorithm of our proposed method is described in Figure B-1.

At the beginning of our algorithm, the system is provided with only positive examples and unlabeled examples. The positive examples are collected by augmenting queries with hypernyms or associative words by human in an ad-hoc manner using full text search. At each iteration of active learning, our method first generates pseudo negative examples (line 10 in Figure B-1). As shown in Figure B-2, we train the naive Bayes classifier using all the unlabeled examples as negative examples, and we predict the word sense of each unlabeled example using the classifier. If the prediction score *c(d, psdNeg)* as defined in the equation (1) is more than a threshold value τ, which means the example *d* is very likely to be negative, it is considered as a pseudo negative example (line 7 in Figure B-2).

$$c(d, psdNeg) = c(d, neg) - c(d, pos) \qquad (1)$$

```
# Definition
Γ(P, N): WSD system trained on P as positive examples and
                                        N as negative examples.
# Input
T ← training dataset including unlabeled dataset
# Initialization
P ←  positive training dataset by full text search on T
N ← ∅ (initial negative training dataset)
repeat
   generate pseudo negative examples PN (see Figure B-2)
   Γ ← Γ(P, PN + N):  newly built WSD system
   #  building training dataset by Active Learning
   cmin ← ∞
   foreach d ∈ ( T – P – N ) do
       classify d by WSD system Γ
       s(d) ← word sense prediction for d using Γ
       c(d, s(d)) ← the confidence score of prediction of d
       if  c(d, s(d))  <  cmin  then  cmin ← c(d);  d min ← d  end
   end
   provide correct sense s for d min  by human
   if  s is positive  then  add d min to P
   else  add d min to N
 until  the number of labeled data reaches the pre-defined counts
```

Figure B-1. Active learning with pseudo negative examples

```
foreach  d ∈ (T – P – N) do
    classify d by WSD system Γ(P, T – P)
    c(d, pos) ← the confidence score that d is predicted as positive by equation (2)
    c(d, neg) ← the confidence score that d is predicted as negative by equation (2)
    c(d, psdNeg) = c(d, neg) – c(d, pos)
        (the confidence score that d is predicted as pseudo negative)
    PN ← d ∈ {(T – P – N) | s(d) = neg ∧ c(d, psdNeg) ≧ τ}
                    (PN is pseudo negative dataset)
end
```

Figure B-2. Selection of pseudo negative examples

Using naive Bayes classifier we can estimate the confidence score c(d, s) that the sense of a data d is predicted as sense s.

$$c(d,s) = \log p(s) + \sum_{j=1}^{j=1} p(f_j|s), \qquad (2)$$

where a data d has features f_1, f_2, ..., f_n. The sense s is positive (pos) when it is the target meaning in a web mining application, otherwise s is negative (neg).

Then, provided with the positive and negative (including the pseudo negative) examples, we again calculate the confidence score for each unlabeled examples (line 11 in Figure B-1). We use the same score function for calculating the confidence score with one for the naive Bayes classifiers.

In building training dataset by active learning, we use *uncertain sampling* (Chan and Ng 2007). As shown in the lines 13-19 of Figure B-1, these steps select the most "uncertain" example $d_{\min}$ that is predicted with the lowest confidence by WSD system Γ. Then, the correct sense for the most uncertain example $d_{\min}$ is provided by human and added to the positive dataset or the negative dataset according to the correct sense of d by human (lines 20-22 in Figure B-1).

The above steps are repeated until all the unlabeled examples are annotated.

As features in equation (2), we use information of the surrounding words of the target word for disambiguation (the target word) as features. Note that we call the phrase (*bunsetsu* for Japanese) which includes the target word the target phrase, and we also call the sentence which includes the target word the target sentence. We use the

following features in this method: (i) the words in depending and governing phrases of the target phrase, (ii) the previous and the next words of the target word within the target phrase, and (iii) the words in local three sentences, i.e., the target sentence and the previous sentence of the target sentence and the next sentence of the target sentence.

We demonstrated that our proposed active learning method achieved effective WSD even starting with no negative examples by four data examples in Japanese blog: (a) "Wega" (product name of TV); (b) "Loft" (store name); (c) "Honda" (personal name of football player); and "Tsubaki" (product name of shampoo). The data size for these example data are: 5,372 for (a); 1,582 for (b); 2,100 for (c); and 2,022 for (d), respectively. And the numbers of ambiguous senses for these example data are: 11 for (a); 5 for (b); 25 for (c); and 6 for (d), respectively. If we assume the upper limit of WSD accuracy is the case we use all training data with labels, at the beginning, the difference between the proposed method and the upper limit are approximately 10 points for (a); 39 points for (b); 10 points for (c) and 10 points for (d). However, the proposed method with 50 percent of the training data obtained, the difference from the upper limit becomes within approximately only two points for all example data. At this points, the proposed method also show higher WSD accuracy compared with one by an simple active learning selecting all unlabeled data as pseudo negative examples.

We also proposed a combination of active learning and semi-supervised learning method, using pseudo negative examples estimated by the classifier trained with positive and unlabeled examples (Imamura, Takayama et al. 2009). This revised version of our proposed method uses EM algorithm with unlabeled data integrated into active learning (McCallum and Nigam 1998; Nigam et al. 2000). We found that there is really not much difference in accuracy between the original proposed method and the revised one in experiments.

The related work for this appendix is divided into two categories, that is studies for WSD and studies for creation of training data.

The approaches of training for WSD are classified into three types, which are supervised, unsupervised and dictionary-based method (Manning and Schütze, 1999). Supervised WSD methods such as Gale et al. (1992) and Brown et al. (1991) are based on only labeled training set. Unsupervised WSD related methods such as Schütze (1998), Lin and Pantel (2002), Pantel and Lin (2002) do not use any labeling data, so these methods did not provide a function to improve the accuracy by adding labeled training data. Dictionary-based WSD such as Shirai and Yagi (2004) is based on lexical resources. Target words in Web mining are often proper nouns, and their word senses are rarely listed in hand-crafted lexicon, therefore, dictionary-based approach is not fit for our target problem.

There are two types of approaches of creation of training data: active learning based methods; and methods using positive and unlabeled data. Chan and Ng (2007) also used active learning approach for WSD for the purpose of domain adaptation. Chen et al. (2006), Zhu and Hovy (2007) studied text classification using active learning in

unbalanced positive and negative examples. However, these previous studies have not treated the case where no negative examples in starting points and did not use pseudo negative examples for efficient active learning. And they have not address active learning for the cases where collecting large amount of negative examples is very costly. Liu el al. (2003), Li and Liu (2005), Li et al. (2007) and Zhu (2007) presented a method to build text classifiers using positive and unlabeled examples. But they did not address active learning approach.

To the best of our knowledge, before we proposed our method, no work have employed active learning using pseudo negative examples that are estimated by training classifiers using positive and unlabeled examples. Our proposed method effectively uses pseudo-negative examples by estimating confidence score which represents degree of ”negativeness” for unlabeled examples, even in a situation only a portion of positive data is fixed in the training data at the starting point of learning.

Acknowledgments

First, I would like to express my sincere appreciation to Professor Yoichi Tomiura for supervising this dissertation.

I would like to thank the member of my dissertation committee: Professor Kumiko Tanaka-Ishii and Professor Shuji Kijima. I am also grateful to the member of my advisory committee: Professor Ryuzo Hasegawa, Professor Tunenori Mine, and Professor Kenji Yoshimura (Fukuoka University).

For the work described through Chapter 2 to Chapter 4, I would like express my gratitude to Professor Douglas W. Oard (University of Maryland), Professor Kenneth R. Fleischmann (The University of Texas at Austin), Professor An-Shou Cheng (National Sun Yat-sen University), and Professor Emi Ishita. Thanks go to Mr. Zheng Wang for discussion. Thanks also go to Jordan Boyd-Graber (University of Colorado Boulder, previously University of Maryland) and the participants of "paper clinic" at UMIACS (Institute for Advanced Computer Studies, University of Maryland) held in January 2014.

For the work described in Appendix B, I would like express my gratitude to Professor Masaru Kitsuregawa (Director General of National Institute of Informatics and The University of Tokyo), Professor Masashi Toyoda (The University of Tokyo), Professor Nobuhiro Kaji (The University of Tokyo), and Dr. Makoto Imamura (Mitsubishi Electric Corporation).

I would like to express my appreciation to co-supervisors of my master thesis, the late Professor Sho Yoshida (former president of Kyushu Institute of Design) and Professor Emeritus Toru Hitaka for their guidance on natural language processing.

I would like to express my appreciation to Professor Emeritus Stanley Peters for my visiting research at the Center for the Study of Language and Information (CSLI), Stanford University during 1997-1998. Thanks also go to his Infomap project members at that time: Professor Genichiro Kikui (Okayama Prefectural University), Professor Hinrich Schütze (University of Munich), Dr. Raymond Flournoy, Professor Stefan Kaufmann (University of Connecticut), Professor Tatsunori Mori (Yokohama National University).

I am indebted to Mr. Masahiro Hirano, the former president of Tokuyama College of Technology, and Dr. Naoki Inoue, the current president of National Institute of Technology, Tokuyama College, for allowing me to conduct my doctoral thesis research.

I thank my former group managers and team leaders at Mitsubishi Electric Corporation, Dr. Takashi Dasai, Mr. Fuyuki Maruyama, Mr. Katsushi Suzuki, Dr. Makoto Imamura, Mr. Yasuhiro Okada, and Mr. Takenori Kawamata.

I also thank my former colleagues of Mitsubishi Electric Corporation and my current colleagues of National Institute of Technology, Tokuyama College.

This dissertation work was supported in part by Japan grant-in-aid for scientific research (B) 25280118, and in part by grant-in-aid for special purposes from the president of National Institute of Technology, Tokuyama College (Tokuyama College of Technology).

The work described in Appendix B was supported in part by "leading-edge storage technology and web analysis technology" as a theme of "Comprehensive Development of e-Society Foundation Software", the leading project of Ministry of Education, Culture, Sports, Science and Technology.

Finally, I wish to thank my friends, my family grew me up, and last but not least my own family Fumiko and Satoko.

References

Artstein, R. and Poesio, M. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555-596.

Bengston, D. N., Webb T. J. and Fan, D. P. 2004. Shifting Forest Value Orientations in the United States, 1980-2001: A computer content analysis. *Environmental Values,* 13(3):373-392.

Blei, D. M., Ng, A. Y. and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*. 3(4-5):993-1022.

Blei, D. M. and Mcauliffe, J. D. 2007. Supervised Topic Models. *Neural Information Processing Systems*, 17:537–544.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J. and Mercer, R. L. 1991. Wordsense disambiguation using statistical methods. *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics(ACL'91)*, pp. 264 -270.

Chan, Y. S. and Ng, H. T. 2007. Domain Adaptation with Active Learning for Word Sense Disambiguation. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistic (ACL2007)*, pp. 49-56.

Chen, J., Schein, A., Ungar, L. and Palmer, M. 2006. An Empirical Study of the Behavior of Active Learning for Word Sense Disambiguation, *Proceedings of main conference on Human Language technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL-2006)*, pp. 120-127.

Cheng, A.-S. and Fleischmann, K. R. 2010. Developing a meta-inventory of human values. *Proceedings of the American Society for Information Science and Technology (ASIST2010)*, 47(1):1-10.

Cheng, A.-S. 2012. *Values in the Net neutrality debate: Applying content analysis to testimonies from public hearings*. PhD dissertation. University of Maryland, College Park, MD, USA.

Cheng, A.-S., Fleischmann, K. R., Wang, P., Ishita, E. and Oard, D. W. 2012. The Role of Innovation and Wealth in the Net Neutrality Debate: A Content Analysis of Human Values in Congressional and FCC Hearings. *Journal of the American Society for Information Science and Technology (JASIST)*, 63(7):1360-1373.

Church, K. W. and Hanks, P., 1990. Word Association Norms, Mutual Information, and Lexicography", *Computational Linguistics*, 16(1), pp. 22-29.

Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement,* 20:37-46.

Fleischmann, K. R., Oard, D. W., Cheng, A.-S., Wang, P. and Ishita, E. 2009. Automatic Classification of Human Values: Applying Computational Thinking to Information Ethics. *Proceedings of the 72nd Annual Meeting of the American Society for Information Science and Technology (ASIST 2009)*, Vancouver, BC, Canada.

Fleischmann, K. R. 2014. *Information and Human Values*, Morgan & Claypool.

Fleischmann, K. R., Oard, D., Cheng, A.-S., Boyd-Graber, J., Templeton, T. C., Ishita, E., Koepfler, J. A. and Wallace, W. A. 2012. Content Analysis for Values Elicitation. *Proceeding of the ACM SIGCHI 2012 Conference on Human Factors in Computing Systems, Workshop on Methods for Accounting for Values in Human-Centered Computing*, Austin, TX, USA.

Fleischmann, K. R., Takayama, Y., Cheng, A.-S., Tomiura, Y., Oard, D. W. and Ishita, E. 2015. Thematic Analysis of Words Used to Invoke Values in the Net Neutrality Debate, *Proceedings of iConference 2015*, New Port Beach, CA, USA (to appear).

Friedman, B., Kahn, Jr., P. H. and Borning, A. 2006. Value Sensitive Design and Information Systems. In P. Zhang and D. Galletta. eds., *Human-computer interaction in management information systems: Foundations*, pp. 348-372, New York: M. E. Sharpe.

Gale, W. A., Church, K. W. and Yarowsky, D. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6):415-439.

Griffiths, T. L. and Steyvers, M. 2004. Finding Scientific Topics. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 101(Suppl. 1):5228-5235.

Griffiths, T. L., Steyvers, M., Blei, D. M. and Tenenbaum, J. B. 2005. Integrating Topics and Syntax. *Advances in Neural Information Processing Systems,* 17.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10-18.

Hindle, D. 1990. Noun Classification from Predicate-Argument Structures, *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics (ACL'90)*, pp. 268-275.

Hsieh, H-F. and Shannon, S. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research*, 15(9): 1277-1288.

Imamura, M., Takayama, Y., Kaji, N., Toyoda, M. and Kitsuregawa, M. 2009. A Combination of Active Learning and Semi-supervised Learning Starting with Positive and Unlabeled Examples for Word Sense Disambiguation: An Empirical Study on Japanese Web Search Query, *Proceedings of Joint Conference of the 47th Annual Meeting of Association for Computational Linguistics & the 4th International Joint Conference on Natural Language Processing of Asian Federation of Natural Language Processing (ACL-IJCNLP 2009) Short Papers*, pp. 61-64.

Ishita, E., Oard, D. W., Fleischmann, K. R., Cheng, A.-S. and Templeton, T. C. 2010. Investigating Multi-Label Classification for Human Values. *Proceedings of the American Society for Information Science and Technology (ASIST2010)*, 47(1):1-4.

Japkowicz, N. and Shah, M. 2011. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Joachims, T. 1998. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. LS-8 Report 23, University of Dortmund.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines*. Springer Science+Business Media, New York.

Kahle, L. R., Poulos, B. and Sukhdial, A. 1988. Changes in Social Values in the United States During the Past Decade. *Journal of Advertising Research*, 28:35-41.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. 1983. *Optimization by simulated annealing*, *Science, 220*:671-680.

Kluckhohn, C. 1951. Values and Value-Orientations in the Theory of Action: An exploration in definition and classification. In Parsons T. and Shils E. eds., *Toward a general theory of action*, Harvard University Press.

Landis, J. R. and Koch, G. G. 1977. A One-Way Components of Variance Model for Categorical Data. *Biometrics*, 33:671-679.

Li, X.-L., and Liu, B. 2005. Learning from Positive and Unlabeled Examples with Different Data Distributions. *Proceedings of 16th European Conference on Machine Learning (ECML'05)*, pp. 218–229.

Li, X.-L., Liu. B. and Ng, S.-K. 2007. Learning to Identify Unexpected Instances in the Test Set. *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pp. 2802-2807.

Lin, D. and Pantel, P. 2002. Concept Discovery from Text, *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 577-583.

Liu, B., Dai, Y., Li, X., Lee, W. S. and Yu, P. S. 2003. Building Text Classifiers Using Positive and Unlabeled Examples. *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 179-187.

Liu, B., Hu. M. and Cheng, J., 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of the 14th International World Wide Web Conference on Data Mining*, pp. 342-351.

Liu, B. 2011. Opinion Mining and Sentiment Analysis. In *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Data-Centric Systems and Applications*, 459-526, Springer-Verlag Berlin Heidelberg.

McCallum, A. and Nigam, K. 1998. Employing EM and Pool-Based Active Learning for Text Classification. *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*, pp. 350-358.

Manning, C. D., Raghavan, P. and Schütze, H. 2008. *Introduction to Information Retrieval*, Cambridge University.

Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing,* MIT Press.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* , 38(11): 39-41.

Minka, T. and Winn, J. 2008. Gates: A Graphical Notation for Mixture Models. MSR-TR-2008-185, Microsoft Research.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. 2000. Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, 39:103-134.

Noreen, E. W. 1989. *Computer Intensive Methods for Testing Hypotheses : An Introduction*, A Wiley-Interscience Publication.

Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval,* 2(1-2): 1-135.

Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. pp. 613-619.

Porter, M. F. 1980. An Algorithm for Suffix Stripping. In Spärck Jones, K. and Willett, P. eds. (1997), *Readings in Information Retrieval*, pp. 313-316, Morgan Kaufmann.

Ramage, D., Hall, D. , Nallapati, R., and Manning, C. D. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP2009)*, pp. 248–256.

Rokeach, M. 1973. *The Nature of Human Values*, New York: Free Press.

Salton, G. M. and McGill, M. J. 1988. The SMART and SIRE Experimental Retrieval Systems. In Spärck Jones, K. and Willett, P. eds. (1997), *Readings in Information Retrieval*, pp. 381-399, Morgan Kaufmann.

Scheible, C. 2014. Supervised and Semi-Supervised Statistical Models for Word-Based Sentiment Analysis, PhD dissertation, The University of Stuttgart.

Schmid, H. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, USA.

Schütze, H. 1998. Automatic Word Sense Discrimination, *Computational Linguistics*, 24(1): 97-124.

Schwartz, S. H. 1994. Are There Universal Aspects in the Structure and Contents of Human Values? *Journal of Social Issues*, 50(4):19-45.

Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1-47.

Shirai, K. and Yagi, T. 2004. Learning a Robust Word Sense Disambiguation Model using Hypernyms in Definition Sentences, *Proceedings of the 20th International Conference on Computational Linguistics (ACL2004)*, pp. 917-923.

Smucker, M. D., Allan, J. and Carterette, B. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation, *Proceedings of the 16th ACM Conference on information and knowledge management (ACM CIKM '07)*, pp. 623-632.

Takayama, Y., Imamura, M., Kaji, N., Toyoda, M. and Kitsuregawa, M. 2009. Active Learning with Pseudo Negative Examples for Word Sense Disambiguation in Web Mining, *IPSJ Transactions on Databases (TOD)*, 2(2): 1-9.

Takayama, Y., Tomiura, Y., Ishita, E., Oard, D. W., Fleischmann, K. and Cheng, A.-S. 2014. A Word-Scale Probabilistic Latent Variable Model for Detecting Human Values, *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (ACM CIKM '14)*, pp. 1489-1498.

Takayama, Y., Tomiura, Y., Ishita, E., Wang, Z., Oard, D. W., Fleischmann, K. R. and Cheng, A.-S. 2013. Improving Automatic Sentence-Level Annotation of Human Values Using Augmented Feature Vectors. *Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING 2013)*, in CD-ROM.

Tausczik, Y. R. and Pennebaker, J. W. 2010. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods, *Journal of Language and Social Psychology*, Vol. 29, No. 1, pp. 24–54 (2010).

Templeton, T. C., Fleischmann, K. R. and Boyd-Graber, J. 2011. Simulating Audiences: Automating Analysis of Values, Attitudes, and Sentiment. *Proceedings of the Third IEEE International Conference on Social Computing*, Boston, MA.

Verplanken, B. and Holland, R. W. 2002. Motivated decision making: effects of activation and self-centrality of values on choices and behavior. *Journal of Personality and Social Psychology*, 82(3):434-447.

Wiebe, J. M. 1994. Tracking Point of View in Narrative. *Computational Linguistics*, 20(2):233-287.

Zhu, J. and Hovy, E. 2007. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem, *Proceedings 2007 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), pp. 783-790.

Zhu, X. 2007. A Semi-Supervised Learning Literature Survey, Computer Science TR 1530, University of Wisconsin.

List of Publications

Publications related to the thesis

1. Fleischmann, K. R., Takayama, Y., Cheng, A.-S., Tomiura, Y., Oard, D. W. and Ishita, E.: Thematic Analysis of Words that Invoke Values in the Net Neutrality Debate, *Proceedings of the iConference 2015* (to appear) (2015).

2. Takayama, Y., Tomiura, Y. Ishita, E., Oard, D. W., Fleischmann, K. R. and Cheng, A.-S.: A Word-Scale Probabilistic Latent Variable Model for Detecting Human Values, *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (ACM CIKM 2014)*, pp.1489-1498 (2014).

3. Takayama, Y., Tomiura, Y., Ishita, E., Wang, Z., Oard, D. W., Fleischmann, K. R. and Cheng, A.-S.: Improving Automatic Sentence-Level Annotation of Human Values Using Augmented Feature Vectors, *Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING 2013)*, in CD-ROM (2013).

4. Imamura, M., Takayama, Y., Kaji, N., Toyoda, M. and Kitsuregawa, M.: A Combination of Active Learning and Semi-supervised Learning Starting with Positive and Unlabeled Examples for Word Sense Disambiguation: An Empirical Study on Japanese Web Search Query, *Proceedings of Joint Conference of the 47th Annual Meeting of Association for Computational Linguistics & the 4th International Joint Conference on Natural Language Processing of Asian Federation of Natural Language (ACL-IJCNLP 2009) Conference Short Papers*, pp. 61-64 (2009).

5. Takayama., Y., Imamura, M., Kaji, N., Toyoda, M. and Kitsuregawa, M.: Active Learning with Pseudo Negative Examples for Word Sense Disambiguation in Web Mining (in Japanese), *IPSJ Transaction on Databases*, Vol. 2, No. 2, pp. 1-9 (2009).

Other Publications

1. Imamura, M., Takayama, Y., Akiyoshi, M. and Komoda, N.: Acquisition of Term Knowledge from Operating Manuals for Information Equipments by Using the Structure of Headline Sentences, (translated from IEEJ Trans. EIS, Vol. 128-C, No. 12), *Electronics and Communications in Japan*, Vol. 93, No. 11, pp. 8-19 (2010).

2. Aikawa, T., Kawamata, T., Mikami, T. and Takayama, Y: A Speed-up Technique of Keyword Suggestion for Document Retrieval in Information Devices by using Augmented Double-array, *Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING 2009)*, pp. 296-301 (2009).

3. Imamura, M., Takayama, Y., Akiyoshi, M. and Komoda, N.: An Acquisition Method on Term Knowledge from Operating Manuals for Information Equipments by Using the Structure of Headline Sentences (in Japanese), *Journal of The Institute of Electrical Engineers of Japan (IEEJ), Transaction on EIS*, Vol. 128-C, No. 12, pp.1833-1841 (2008).

4. Imamura, M., Takayama, Y., Akiyoshi, M. and Komoda, N.: Term Knowledge Acquisition Using the Structure of Headline Sentences From Information Equipments Operating Manuals, *Proceedings of the 3rd International Conference on Software and Data Technologies (ICSOFT 2008)*, pp. 301-308 (2008).

5. Imamura, M., Takayama, Y., Okada, Y. and Komoda, N.: A Document Rules Description Language based on Feature Logic for XML Document Exchange, *Proceedings of the International Conference on E-business and Telecommunications (ICE-B 2007)*, pp. 71-77 (2007).

6. Takayama, Y., Flournoy, R. S., Kaufmann, S. and Peters, S.: Information Retrieval based on Domain-Specific Word Associations, *Proceedings of the 4th Conference of the Pacific Association for Computational Linguistics (PACLING '99)*, pp.155-16 (1999).

7. Flournoy, R. S., Ginstrom, R., Imai, K., Kaufmann, S., Kikui, G., Peters, S., Schütze, H. and Takayama, Y.: Personalization and users' semantic expectations, *Proceedings of the ACM SIGIR'98 Post-Conference Workshop on Query Input and User Expectations*, pp. 31-35 (1998).

8. Takayama, Y. Ito, T. and Suzuki, K.: An Application of Natural Language Case-based Retrieval to Product Design Support, *Proceedings of the Workshop on Practical Use of Case-Based Reasoning, The 15th International Joint Conference on Artificial Intelligence (IJCAI '97)*, pp. 31-77 (1997).

9. Isahara, H., Uchino, H., Ogino, S., Okunishi, T., Kinoshita, S., Shibata, S., Sugio, T., Takayama, Y., Doi, S., Nagano, T., Narita, M. and Nomura, H.: Technical Evaluation of MT Systems from the Developer's Point of View: Exploiting Test-Sets for Quality Evaluation (in Japanese), *Journal of Natural Language Processing*, Vol. 3, No. 3 (1996).

10. Isahara, H., Uchino, H., Ogino, S., Okunishi, T., Kinoshita, S., Shibata, S., Sugio, T., Takayama, Y., Doi, S., Nagano, T. Narita, M. and Nomura, H.: Technical evaluation of MT systems from the developer's point of view: exploiting test-sets for quality evaluation, *Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA-94)*, pp. 126-133 (1994).

11. Takayama, Y., Suzuki, K., Maruyama, F. and Dasai, T: A Prolog-based Practical Machine Translation System MELTRAN-J/E, *Proceedings of the 2nd Japan-Australia Joint Symposium on Natural Language Processing*, (JAJSNLP'91), pp. 225-233 (1991).