Improving Automatic Sentence-Level Annotation of Human Values Using Augmented Feature Vectors

Yasuhiro Takayama,[1,2] Yoichi Tomiura,[2] Emi Ishita,[2] Zheng Wang,[2,*] Douglas W. Oard,[3]
Kenneth R. Fleischmann[4] and An-Shou Cheng[5]
[1]*Tokyama College of Technology,* [2]*Kyushu University,* [3]*University of Maryland,*
[4]*University of Texas at Austin,* [5]*National Sun Yat-sen University*
takayama@tokuyama.ac.jp, {tom@inf, ishita.emi.982@m}.kyushu-u.ac.jp, kendovivi@hotmail.com, oard@umd.edu,
kfleisch@ischool.utexas.edu, ascheng@mail.nsysu.edu.tw

Abstract

This paper describes an effort to improve identification of human values that are directly or indirectly invoked within the prepared statements of witnesses before legislative and regulatory hearings. We automatically code human values at the sentence level using supervised machine learning techniques trained on a few thousand annotated sentences. 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 about a 6% relative improvement in F_1 using a Support Vector Machine classifier.

Key Words - text classification, human values, computational social science

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 [1, 2]. 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 [3]. In this paper, 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 [4], we 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 [3], we need to improve our ability to effectively analyze larger data corpora. In this paper, 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 our previous work.

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 16.5 words), 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”). 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 paper also reports on an experiment comparing multiple human annotators and the classifier to explore

[*] Currently with Knowledge Engineering & Integration Development Ltd.

the possibility of replacing human annotators with our classifier.

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

2. The Meta-Inventory of Human Values

The value categories for this study were selected from the Meta-Inventory of Human Values (MIHV) [6]. 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 [7]. 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 [8]. Four rounds of annotation were conducted to refine the annotation guidelines. For each round, four documents were randomly selected from the corpus for annotation by the seventh author of this paper and two independent annotators. Cohen's Kappa [9] was used to characterize inter-annotator agreement, and Landis & Koch's guidelines [10] were used to interpret the Kappa values, as is common in computational linguistics and other domains [11].

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 categories were then used by the seventh author of this paper to annotate the entire corpus. Twenty documents from the corpus were annotated by a second annotator. The Kappa values of the six value categories for these 20 documents are shown in Table 2.

3. Test Collection

The corpus for this study was created from written opening statements and testimonies prepared for and delivered at public hearings held by the U.S. Congress and the U.S. Federal Communications Commission (FCC). 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 paper. 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* [7]). The ultimate goal of our work is to be able to replicate similar experiments at a larger scale. All 9,890 sentences in each of 102 documents were manually annotated by one of the authors.[1] Table 1 shows some examples. 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.

We limited the corpus to sentences of 40 words or less, because sentence boundaries for longer sentences are sometimes different from those of a tokenizer. We removed sentences annotated as section headings then 8,713 sentences remained. Finally we also removed null sentences after eliminating stop words, leaving 8,660 sentences for the experiments described in this paper. The numbers of sentences for each value are shown in Table 2.

Table 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.

Table 2. Prevalence, inter-annotator agreement (kappa)

Value	κ	# doc	sentences (original)	sentences (used)
wealth	0.629	102	3,563	3,156
social order	0.683	102	2,859	2,503
justice	0.420	99	2,641	2,267
freedom	0.620	101	2,431	2,155
innovation	0.715	94	1,147	1,018
honor	0.430	80	352	317

[1] The author who performed the annotation did not participate in the design of the classifiers reported on in this paper.

4. Proposed Approach

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 [5]. 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) [12, 13]. Both are unsupervised, not requiring any human annotation.

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

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

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 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 $Te_i$ ($1 \le i \le N$) consist of word id's
    with score 1, that occurred in each sentence.
    (The suffix $N$ represents $N$-fold cross-validation.)
04: Construct classifier $\Gamma^{(v)}_{\text{base}}$ by learning the training
      data $Tr_1, ..., Tr_N$ for each value ($v$).
05: Classify $Te_1, ..., Te_N$ by the classifier $\Gamma^{(v)}_{\text{base}}$, 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 $ATe_1, ..., ATe_N$.
    $ATr_i = Tr_i$ [+ $Pr_i$] [+$Cr_i$ ] [+$Sr_i$ ] [+$Hr_i$ ],
    $ATe_i = Te_i$ [+ $Pe_i$] [+$Ce_i$ ] [+$Se_i$ ] [+$He_i$ ],
    where + represents the vector concatenation operator
    and [ ] represents optional.
10: Construct classifier $\Gamma^{(v)}_{\text{modified}}$ by learning the
      augmented training data $ATr_1, ..., ATr_N$.
11: Classify the augmented test $ATe_1, ..., ATe_N$ by
      $\Gamma^{(v)}_{\text{modified}}$ and evaluate the result.
```

Figure 1. Classification with augmented feature vectors.

5. Experiments

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

5.1 Preliminary result and Classifier Design

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

(1) Lemmatization using TreeTagger 3.2,[2] to normalize each word to its corresponding WordNet[3] root form.

(2) Stopword removal using the SMART stopword list,[4] adapted for TreeTagger's output.

(3) Stemming by Porter's stemmer [14].

Before focusing on SVM results, we conducted a preliminary experiment comparing k-Nearest Neighbor (kNN), naive Bayes (NB) and SVM classifiers. We used the University of Waikato's Weka toolkit[5] for kNN (with k=1) and NB, and throughout this paper we use TinySVM[6] (with a second-degree polynomial kernel) as our SVM classifier. Table 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 polynomial kernel SVM classifiers for the remainder of this paper.

Table 3. Classifier selection (102-fold doc cross-val).

Classifier	Prec	Recall	F_1
kNN (k = 1)	0.6058	0.3434	0.4350
naive Bayes	0.5260	0.6177	0.6333
SVM	**0.7730**	**0.6510**	**0.7068**

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 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 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.

5.2 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 in Table 4.

Table 4. Human "classifier" effectiveness (20 docs).

Value	Prec	Recall	F_1
wealth	0.7345	0.8711	0.7970
social order	0.7751	0.7588	0.7669
justice	0.6635	0.4638	0.5460
freedom	0.6810	0.7682	0.7220
innovation	0.7644	0.7197	0.7414
honor	0.3950	0.5529	0.4608
average	0.7117	0.7320	0.7217

Comparable results for the baseline classifier, tested on the same 20 documents and trained on the remaining 82 documents, are shown in Table 5. 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*!

Table 5. Baseline classifier (82 train:20 test docs).

Value	Prec	Recall	F_1
wealth	0.8160	0.6536	0.7258
social order	0.7828	0.7468	0.7644
justice	0.6518	0.5691	0.6076
freedom	0.7362	0.7048	0.7201
innovation	0.7800	0.6393	0.7027
honor	0.4800	0.1412	0.2182
average	0.7550	0.6505	0.6988

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.

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

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

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

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

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

5.3 Overall Effect of Expansion

Table 6 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 6. Average (25% train:75% test, 102-doc cross-val).

Method	Prec	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 3 and 6 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 6 with the baseline indicates that augmenting feature vectors using both CP and syn' recovers some of that loss, yielding a 0.035 absolute (and 5.4% relative) 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).

5.3 Per-Category Analysis

Averages can hide important details, so we also report results for each of our six human value categories. Table 7, corresponds to the first line in Table 6; Table 8 corresponds to the second to last line in that table (expansion using CP+syn', which yields gives the best average F_1). Table 9 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 7. Baseline (25% train:75% test, 102-doc cross-val).

Value	Prec	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 8. CP+syn' Classifier (same cond. as Table 6).

Value	Prec	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 9. Relative F_1 improvement (from Table 7 to 8).

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

5.4 Sentence Cross Validation

In prior work, we had reported results for 10-fold cross-validation, using randomly selected sentences [4]. Table 10 shows results for that design (with our present values categories; in our earlier work we had used a different values inventory).

Table 10. Classifier effectiveness (10-fold sent. cross-val).

Method	Prec	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.

6. Conclusion and Future Work

In this paper, 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. The combination of evidence from statistical term associations and lexical evidence for synonyms has been shown to be effective. We have improved very substantially over our previously reported results 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 substantial improvements in the effectiveness of sentence classification can be achieved using expansion.

For this paper, we have adopted a relatively conventional approach to evaluation, measuring the effect of errors on individual errors using F_1. For some types of content analysis in social science, however, counterbalancing errors (one false positive for each false negative) might not affect the conclusions that we draw. This suggests that we may actually wish to minimize bias rather than accuracy, and therefore in future work we plan to also explore measures in which we focus on the bias-variance tradeoff.

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 vast and noisy Web-scale information [3]. That is our ultimate goal.

Acknowledgments

This work was supported in part by NSF IIS-0725459, Japan grant-in-aid for scientific research (C) 22500220 and (B) 25280118.

References

[1] Pang, B., Lee, L., "Opinion mining and sentiment analysis", *Foundations and Trends in Information Retrieval,* 2(1-2), pp. 1-135 (2008).

[2] Liu, B., "Opinion Mining and Sentiment Analysis", *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Data-Centric Systems and Applications*, pp. 459-526, Springer-Verlag Berlin Heidelberg (2011).

[3] Fleischmann, K.R., Cheng, A.-S., Templeton, T.C., Koepfler, J.A., Oard, D.W., Boyd-Graber, J., Ishita, E., Wallace, W.A., "Content Analysis for Values Elicitation", *SIGCHI Workshop on Methods for Accounting for Values in Human-Centered Computing*, Austin, TX (2012).

[4] Ishita, E., Oard, D.W., Fleischmann, K.R., Cheng, A.-S., Templeton, T.C., "Investigating multi-label sentence classification for human values, *73rd Annual Meeting of ASIST,* Pittsburgh, PA (2010).

[5] Sebastiani, F., "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, 34(1), pp. 1-47 (2002).

[6] Cheng, A.-S., Fleischmann, K.R., "Developing a meta-inventory of human values", *73rd Annual Meeting of ASIST*, Pittsburgh, PA (2010).

[7] Cheng, A.-S., Fleischmann, K.R., Wang, P., Ishita, E., Oard, D., "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*, 63(7), pp. 1360-1373 (2012).

[8] Cheng, A.-S., "*Values in the Net Neutrality Debate: Applying Content Analysis to Testimonies from Public Hearings"*, Ph.D. Dissertation, Univ. of Maryland (2012).

[9] Cohen, J., "A coefficient of agreement for nominal scales", *Educational and Psychological Measurement,* 20, pp. 37-46 (1960).

[10] Landis, J.R., Koch, G.G., "A One-Way Components of Variance Model for Categorical Data", *Biometrics*, 33, pp. 671-679 (1977).

[11] Artstein, R., Poesio, M., "Inter-Coder Agreement for Computational Linguistics", *Computational Linguistics*, 34(4), pp. 555-596 (2008).

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

[13] Hindle, D., "Noun Classification from Predicate-Argument Structures", *28th Annual Meeting ACL*, pp. 268-275 (1990).

[14] Porter, M.F., "An algorithm for suffix stripping", in Jones, K. S., Willet, P. (1997), *Readings in Information Retrieval*, pp. 313-316, Morgan Kaufmann (1980).