An Automatic Dictionary Extraction and Annotation Method Using Simulated Annealing for Detecting Human Values

Yasuhiro Takayama
Computer Sci. & Electronic Eng.
Nat'l Inst. of Tech., Tokuyama College
Shunan, Yamaguchi, Japan
takayama@tokuyama.ac.jp

Yoichi Tomiura
Information Sci. & Electrical Eng.
Kyushu University
Nishi-ku, Fukuoka, Japan
tom@inf.kyushu-u.ac.jp

Kenneth R. Fleischmann
School of Information
University of Texas at Austin
Austin, TX, USA
kfleisch@ischool.utexas.edu

An-Shou Cheng
College of Management
National Sun Yat-Sen University
Kaohsiung City, Taiwan
ascheng@mail.nsysu.edu.tw

Douglas W. Oard
iSchool/UMIACS
University of Maryland
College Park, MD, USA
oard@umd.edu

Emi Ishita
Kyushu University Library
Kyushu University
Higashi-ku, Fukuoka, Japan
ishita.emi.982@m.kyushu-u.ac.jp

***Abstract*— This paper studies a method for identifying word unigrams and word bigrams that are associated with one or more human values such as *freedom* or *innovation.* The key idea is to deterministically associate values with word choices, thus permitting values reflected by sentences to be assigned using dictionary lookup. This approach works nearly as well on average as the most accurate existing methods, and at close to the best results that can be achieved by a second human annotator, but the principal contribution of the new method is that the basis for the system's classification decisions are more easily interpreted by social scientists. The new method is based using a Monte Carlo algorithm with simulated annealing to efficiently explore the space for optimal assignments of human values to unigrams and bigrams. Results are reported on an annotated test collection of prepared statements from witnesses at public hearings on the topic of *net neutrality*. The results include accuracy comparisons with the previously reported approach.**

Keywords—Simulated Annealing; Computational Social Science; Human Values

I. Introduction

This paper describes a method for extracting human values dictionary containing word unigrams and bigrams for detecting human values reflected within sentences in prepared statements related to the *net neutrality* debate. Human values such as *freedom* or *justice* are useful to explain people's opinions. The values are found in certain linguistic expressions, typically words and word sequences in texts which writers' appeals are reflected.

Human values have proven to be a useful construct for explaining people's choices. Social scientists have used human values to explain attitudes and behaviors [19]. 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. In the Net neutrality debate, support of net neutrality is positively correlated with valuing *innovation* and negatively correlated with valuing *wealth* [5].

Opinion mining is one of the related research fields to which this paper addressed. In the traditional opinion mining or sentiment analysis, however, the study tends to focus on the polarity of sentiment, which can be positive or negative (or neutral) [15], [17]. Although sentiment analysis or opinion mining are useful in several applications, social scientists have sought to look more deeply for factors that might help to explain or to predict human values in opinions.

Lexical features such as words and word stems have been shown to be a useful basis for detecting human values [13], [21]. We take a step in that direction by constructing a values-dictionary for use in automated detection of human values that are expressed or reflected by the word choice in specific sentences. This values-dictionary in turn can facilitate qualitative analysis for social scientists of the association between human values and word choice by providing cue expressions in an easily used form (specifically, as a list of word or stem unigrams and bigrams).

In this paper, we focus on a method for automatic construction of the values-dictionary and an empirical demonstration that the resulting values-dictionary can be used to achieve classification results that are quite competitive with one of the existing state of the arts. The remainder of this paper is organized as follows. In Section II, we describe background work on human values research, on related social science analysis methods, and on classification methods. Section III then introduces the test collection that we have used. Section IV describes our method for constructing the values dictionary and for using that values dictionary to automatically detect the human values expressed by specific sentences. Section V presents the results of our experiments. Finally, section VI concludes the paper.

II. Related Work

Content analysis is an approach to detecting human values[5], [8], [25]. 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 [11] refer to this combination of human interpretation and a pre-existing coding scheme as a “directed approach”, which limited by the annotation costs along with the size of the collection. Early in the annotation process, social scientists tailor the pre-existing coding scheme for reflecting the characteristics of a collection. Our proposed techniques are intended 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 were able to automate annotation process using text classifiers trained with that data [20].

In the social psychology field, LIWC [23] provides a set of general terminology and category database for psychological meaning of words to analyze text with computers. What distinguishes between LIWC and our proposed values-dictionary based method is that we concentrate on human values and provide automatic dictionary constructing method depending on the targeting corpus to be analyzed.

Several efforts for automatic values classification have been done for some set of sentences. Ishita et al. [12] used *k*-NN to detect human values. Takayama et al. [21] used *k*-NN, naive Bayes, and SVMs (Support Vector Machines) [13] with word features, and then found that SVM gave the best classification effectiveness among them. Thus they improved SVM’s effectiveness by expanding words features with associative words that extracted based on statistical similarity from unlabeled text and/or from hand-crafted thesaurus. In addition, there are several brand new approaches to model natural language text using probabilities [1], [10], [22]. However, these probabilistic approaches tend to focus on improving classification effectiveness in quantitative view. From the point of vies of qualitative analysis (by human), these existing approaches request somewhat skillful technic to master them as a tool and social scientists feel some difficulty with interpretations about numerical scores attached with features.

For example, naive Bayes produces words, which actually do not have the human value, have the value with a certain small probability as output. This is also happened when we use the state-of-the art probabilistic modeling methods like LDA (Latent Dirichlet Allocation) [1], [10] which determines whether the word has the topic probabilistically, so whether the same word has a certain topic (in our case, a human value) or not depends on context (i.e. the sentence) where the word occurs.

The method of simulated annealing [3], [14] which we employed in this paper has traditionally applied to parsing [18] and to word sense disambiguation [7] in natural language processing. The latter approach is similar to ours where they use the method for word-level sense. Our approach is, however, different from it in the respect where our method treat the relationship between sentence-level and word-level values.

When social scientists perform qualitative analysis, it is much easier to interpret relationships between values and sentences if the values pattern of the sentence deterministically depends on just the occurrences of specific words and/or specific word sequences in that sentence. Therefore, we will propose a new method to extracting values-dictionary that is applicable to automatic values estimation for unseen sentences and demonstrate the dictionary is applicable for qualitative analysis by social scientists.

III. Test Collection

The section briefly describes a test collection we use in this paper. The test collection for experiments is available at http://nlp.inf.kyushu-u.ac.jp/values.html. This test collection was originally developed by Cheng [6]. The collection includes 102 written prepared statements from public hearings held by the U. S. Congress and Federal Communications Commission (FCC) on net neutrality. The sentences annotated with some of six human values: *freedom*, *honor*, *innovation*, *justice*, *social order*, *wealth*. These human values are selected from the meta-inventory of human values [4]. Actual test corpus contains 8,660 sentences in 102 documents were stemmed by the Porter stemmer. The average sentence length of our corpus is only 10.3 word stems.

IV. Proposed method for values-dictionary extraction and automatic annotation

Our proposed method is inspired by a Monte Carlo algorithm [16], which is more general than popular MCMC (Markov Chain Monte Carlo) algorithms such as Metropolis-Hastings or Gibbs sampling. We adopt a bit pattern representation for multiple human values. Then we apply a simulated annealing method [3], [14] to assign the pseudo optimal human values pattern for the word unigrams and bigrams. Our proposed algorithm optimizes the values assignment to maximize F_1 measure which is the harmonic mean of precision and recall.

A. Preparation and Notation

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 $\mathbf{w}$ 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, i.e. *freedom*, *honor*, *innovation*, *justice*, *social order*, and *wealth*, respectively. The pattern 000000 means that the corresponding sentence does not have any values.

The human values for a word w (word unigram w) are denoted by $v(w)$. And the values for a word w_2 under the influence of its previous word w_1, i.e., the values for a word bigram (w_1 w_2) are denoted by $v(w_2 ; w_1)$. A *values-dictionary* that we are constructing is a collection of $v(w)$ and $v(w_2 ; w_1)$ for all words and word bigrams. We restrict $v(w)$ and $v(w_2 ; w_1)$ to be an element in χ, where $\chi = \{000000, 000001, 000010, 000011, 000100, ... , 110000\}$. The cardinality of χ is 22.

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, the model without this constraint shows no further improvement.

We define the *word-level values* that are represented by the word sequence $w_{n-1}\ w_n$ within a sentence $\mathbf{w} = (w_1, w_2, ..., w_N)$ using the values-dictionary as follows.

$$x_n = \begin{cases} v(w_n; w_{n-1}) \ ; \ v(w_n; w_{n-1}) \neq 000000 \\ \quad v(w_n) \quad ; \ v(w_n; w_{n-1}) = 000000 \end{cases} \qquad (1)$$

That is, we give a preference to the values $v(w_n\ ;\ w_{n-1})$ for the word position n, which takes account of wider scope over the values $v(w_n)$.

The sentence-level values for a sentence $\mathbf{w}$ is calculated by the following formula:

$$x_1 \oplus x_2 \oplus \cdots \oplus x_N , \qquad (2)$$

where the symbol $\oplus$ is a logical bitwise OR operator.

We can treat the construction of a values-dictionary, that is, the calculations of $v(w)$ and $v(w_2\ ;\ w_1)$ for all words and word bigrams, as a combinatorial optimization problem for text data annotated with sentence-level values. The objective function is interpreted by F_1 measure for estimating the sentence-level values based on the above formula (2). Finding an optimal solution is, however, quite difficult because the possible values for $v(w)$ and $v(w_2\ ;\ w_1)$ are both 22 kinds and the number of distinct words is several thousands.

Therefore, we will find an approximate solution for this optimization problem by a probabilistic search method. To make effectiveness of the approximate solution better, we limit the number of possible values for $v(w)$ and $v(w_2\ ;\ w_1)$. For this purpose, by preliminary manual analysis of our test corpus, we assume the following properties about the relationship between words and their values:

(a) Most words do not have any values, only a few words serve as values indicator.

(b) Some of two-word sequences also have values as the pair of words.

From the above property (a), a word w is less likely to have any values. On the other hand, if a word w has a value k, a sentence where the word w occurs must have the value k unless (w', w) has the values other than k. (the word w' is the previous word of the word w.) However, the latter might be a rare case by the above property (b), thus the most sentences where the word w occurs have a value k, when the word w has the value k. Therefore, we can assume that the word w is likely to have a value k if a certain percentage of sentences in the corpus, where the word w occurs, have the value k. Thus we limit the possible values for $v(w)$, i.e. search space, as follows.

$$[v(w)]_k = \begin{cases} 0 \ or \ 1 \ ; & \dfrac{ns(w,k)}{ns(w)} \geq \alpha, \text{and } ns(w) \geq FW \\ \quad 0 \ ; & \qquad otherwise \end{cases},$$

where $[b]_k$ represents the k-th bit in the bit sequence b. The variable $ns(w)$ is the number of sentences where the single word w occurs. And $ns(w, k)$ is the number of sentences with the value k, where the word w occurs. The meta-parameter α is the minimum ratio. The meta-parameter FW is the minimum frequency of sentences to make the observed counts reliable. The maximum possible values for a word w, denoted by $maxV(w)$, is as follows.

$$[maxV(w)]_k = \begin{cases} 1 \ ; & \frac{ns(w,k)}{ns(w)} \geq \alpha, \text{ and } ns(w) \geq FW \\ 0 \ ; & \qquad otherwise \end{cases}$$

In a similar way, from the above property (a), the word w_2 is less likely to have any values if influence of the previous word w_1 is taken into account. If the word w_2 has the value k under influence of the previous word w_1, all sentences where the word sequence $w_1\ w_2$ occur have the value k. We do not assume this always holds due to noises from influence by the context, thus we also limit the possible values for $v(w_2;\ w_1)$ as follows.

$$[v(w_2; w_1)]_k = \begin{cases} 0 \ or \ 1 \ ; & \dfrac{ns(w_1, w_2, k)}{ns(w_1, w_2)} \geq \alpha, \text{and } ns(w_1, w_2) \geq FWW \\ \quad 0 \ ; & \qquad otherwise \end{cases},$$

where $ns(w_1, w_2)$ is the number of sentences where the bigram (w_1, w_2) occurs. And $ns((w_1, w_2, k)$ is the number of sentences with the value k, where the bigram (w_1, w_2) occurs. FWW is the minimum frequency of sentences to make the observed bigram counts reliable. The maximum possible values for a word bigram (w_1, w_2), denoted by $maxV(w_2; w_1)$, is as follows.

$$[maxV(w_2; w_1)]_k = \begin{cases} 1 \ ; & \frac{ns(w_2; w_1, k)}{ns(w_2; w_1)} \geq \alpha, \text{and } ns(w_1, w_2) \geq FWW \\ 0 \ ; & \qquad otherwise \end{cases}.$$

B. Algorithm for extracting values-dictionary and estimation of sentence-level values

In this subsection, we describe the values-dictionary extraction algorithm and its application to estimation of sentence-level values.

1) Values-dictionary Extraction

Our proposed algorithm for extracting human values is described in Figure 1. First, this algorithm starts from a reasonable assignment (the values we see most often with each word unigrams and bigrams in the annotated training sentences), i.e. $maxV(w)$ and $maxV(w_2;w_1)$ at the line (1). Then the algorithm iteratively tries small variations of values assignment until iterations reached a certain big number (no further improvement of F_1 might be found) between lines from (2) to (21).

The trial seeking for optimal values assignments in search space for each bigram (w_1, w_2) is done by a small probability at the line (6). The same as applied for each unigram w at the line

(19). The assignment stored in $v(w)$ or $v(w_2\ ;\ w_1)$ denoted by the status *C*. The new assignment (the status *C'*) is selected to modify only one bit for a bigram from the previous assignment by the function gen_new_status described in Figure 2, which is called at the line (8). A new assignment is searched within neighborhood of the previous assignment by proximate optimality property. The closed test at (7) and (9) means a calculation of F_1 only for training dataset, by estimating with all $v(w_2;\ w_1)$ using formula (2) only for bigrams. At the line (19), this estimation is done with all $v(w)$ for unigrams. The improvement rate ΔE between F_1 for the current status *C* and F_1 for a new status *C'* is calculated at the line (10).

If F_1 for *C'* is improved at the line (11), the new assignment for $v(w_2; w_1)$ or $v(w)$ is adopted at the line (12). Otherwise, even in the case where F_1 for *C'* is not improved, the new assignment is also adopted by small probability. The latter case, the new status *C'* is adopted only if the number of values is decremented at the line (16). At the line (20), the possibility of the chance of modification in the case of F_1 is not improved gradually becomes smaller. After terminating this algorithm, the values assignments stored in $v(w)$ and $v(w_2; w_1)$, for all unigrams and bigrams with one or more values, are extracted as the entries of the values-dictionary.

2) *Estimation of sentence-level values*

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 or two-word sequence in a sentence to be estimated the sentence-level values, if it matches an entry of the values-dictionary, the word (sequence) is assigned the values which the entry has. When the both a word and a two-word sequence are matched with entries in the values-dictionary, the values assignment is calculated by the equation (1). After the word-level values assignment process is finished, the sentence-level values are calculated by the formula (2).

This sentence-level values estimation process is quite deterministic; therefore, it is readily traceable even by human. By this traceability and understandability of estimation process lead us this method and values-dictionary into creation of annotation manual for novice annotator of human values to train to acquire what human values are and to develop the sense of stable annotation which is required for content analysis.

V. Proposed Method for Values-Dictionary Extraction and Automatic Annotation

In this section, we first describe our experiment design. We then describe some examples of the resulting values-dictionary, we report classifier effectiveness, and we compare our results with both human annotation and the state-of-the-art automated classification.

```
(1)  Calculate the maximum possible values $maxV(w)$ for the
     word $w$ and $maxV(w_2;w_1)$ for the bigram $(w_1,\ w_2)$.
(2)  $TC \leftarrow 1.0$
(3)  Repeat
(4)   foreach bigram $(w_1,\ w_2)$, apply the following
(5)    Generate a uniform random number $r_1$.
(6)    if ( $r_1 < smallProb$ (= 0.001) ) {
(7)      $oldF_1 \leftarrow$ closed test for the current status C.
(8)      $C' \leftarrow$ gen_new_status($w_2;\ w_1$).
(9)      $newF_1 \leftarrow$ closed test for a new status C' .
(10)      $\Delta E \leftarrow 1/(1+ newF_1) - 1/(1+ oldF_1)$ .
(11)      if ($\Delta E < 0$) // $newF_1$ is higher than $oldF_1$
(12)        Set the status C as C' .
(13)     else {
(14)       Generate a uniform random number $r_2$.
(15)        if ( $r_2 < exp\ (-\Delta E\ /TC\ )$ )
(16)         Set the status C' to C if the number of values
             of C' is fewer than C's.
(17)     }
(18)  }
(19)  foreach unigram $w$, apply the same as (5)-(18).
(20)  $TC \leftarrow TC * 0.999$.
(21) Until loop-counter reaches NITER (=100,000)
```

Figure 1. Algorithm for values-dictionary construction using Simulated Annealing

```
gen_new_status ($w$):
  (1) Select assignment whether to increase or to decrease the
      number of the values for the word by the equal
      probability (i.e. probability is 0.5, respectively).
  (2) In increment, choose a $k$ where the $[maxV\ (w)]_k$ is 1 and
      $[v(w)]_k$ is 0 in equal probability, then set $[v(w)]_k$ to 1.
  (3) In decrement, choose the a $k$ where $[v(w)]_k$ is 1 in equal
      probability, then set $[v(w)]_k$ to 0.
   Note. The number of values assigned for a word must be
      less than or equal to 2.
```

Figure 2. A generating function of a new status of values assignment The values-dictionary is a global variable for this function. When this function is applied to a bigram instead of a unigram, read *w* as (w_2; w_1).

A. *Experiment Design*

We use 102-fold document-scale cross-validation. 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 100 documents for development training and one held-out document for development testing. 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-parameter α that yield the best F_1, sweeping the meta-parameters α across 0.5, 0.55, and 0.6; *FW* across 4, 5, 6; and *FWW* across 8, 9, 10. These ranges were determined from preliminary experiments using a portion of the same test collection. Once the parameter choices are made for a particular fold, training is conducted on the full 101-document training set using the best α, and the resulting model is then used to classify the sentences in the single-document test set.

For the baseline, we also apply 102-fold document-scale cross-validation for SVM[1)] either with only unigram features or with unigram and bigram features. We use a 2nd-degree polynomial kernel for the SVM with unigram features and a linear kernel for the SVM with unigram and bigram features, that choice of kernels was made using side experiments on the same test collection. To determine the frequency threshold η for bigram features (used if frequency ≥ η, a meta-parameter for SVM, sweeping η across 1 to 10, and ∞ (= without bigram)), we apply the same development testing process described above.

The assignment of values for word unigrams and word bigrams depends on the training set. Therefore the resulting dictionary entries could be different for each fold in the experiment. Thus we merge all resulting dictionary entries with a bitwise OR operation. Although we limit the dictionary entry for each fold to at most two human values for any unigram or bigram, this merging process can sometimes generate values-dictionary entries that generate more than two values.

B. Result

1) Extracted human values-dictionary

Our proposed method constructs unigrams and bigram dictionaries. The number of unigram stems with values is 461 (33 word stems have multiple values) and the number of stem bigrams is 360 (76 stem bigrams have multiple values). Table 1 shows the entries from the resulting values-dictionary. For each value, the upper rows show the most frequent stem unigrams and the lower rows show the most frequent stem bigrams. The numbers are the corresponding sentence counts. These counts decay quickly, with only a small handful of unigrams and bigrams accounting for the vast majority of the value assignments. We can see the value name "innov(ation)" are included in Table 1, and that it is good cue for detecting the value which they express.

Table 1. Examples of values-dictionary entries

Value	word stem unigram and bigram dict. entries
wealth	market :521 manag :373 invest: 306
	network manag: 214 manag practic : 58
s-order	fcc : 559 regul : 418 rule : 387
	network neutral: 244 net neutral : 205
justice	block:195 discrimin:182 nondiscrimin: 102
	market power : 106 network owner : 105
freedom	competit:597 open : 319 choic : 240
	open internet: 106 consum choc : 69
innov	**innov** : 411 creative : 28 evoluv : 40
	invest **innov** : 24 **innov** competit :20
honor	voter : 10 mission : 10 orlean : 8
	nation larg : 12 trade assoc : 8

2) Classification Effectiveness

Table 2 shows the classifier effectiveness for the proposed method *SA* (stands for *S*imulated *A*nnealing) and for SVM. As for SVM, we found that F_1 = 0.7057 for SVM with unigram and bigram features is lower than F_1 = 0.7068 for SVM with only unigram features in spite of using a meta-parameter for tuning of SVM with unigram and bigram features. Thus, we only show the result for SVM with unigram feature in Table 2 for readability. We can see our proposed *SA* obtained comparable (numerically slightly higher) F_1 against with SVM. As for the comparison between SVM and *SA*. Precision of SVM is much higher than *SA*, while recall of *SA* is much higher than *SA*. Higher recall is useful for human qualitative analysis because human can compensate the misjudgment by machine classifier during verification process for the detail content analysis. What we should emphasize in this result that small simple dictionary with unigrams 461 (only 7% of unique words) and bigrams 360 (only 0.7 % of unique bigrams) achieved quite good effectiveness.

Table 2. Classifier effectiveness for *SA* and SVM.

Value	Precision		Recall		F_1	
	SVM	*SA*	SVM	*SA*	SVM	*SA*
wealth	**0.7859**	0.7135	0.6977	**0.8048**	0.7392	**0.7564**
s-order	**0.8235**	0.7149	0.7587	**0.8554**	0.7898	0.7788
justice	**0.7275**	0.6527	0.5558	**0.5977**	0.6302	0.6240
freedom	**0.7461**	0.7211	0.6654	**0.7151**	0.7035	**0.7181**
innov	**0.8139**	0.7336	0.5629	**0.5923**	0.6655	0.6554
honor	0.4324	**0.4643**	**0.2019**	0.0410	**0.2753**	0.0754
average	0.7730	0.7052	0.6510	**0.7177**	0.7068	**0.7114**

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

Value	Precision		Recall		F_1	
	Human	*SA*	Human	*SA*	Human	*SA*
wealth	0.735	**0.790**	**0.871**	0.784	**0.797**	0.787
s-order	**0.775**	0.696	0.759	**0.897**	0.767	**0.784**
justice	0.664	**0.684**	0.464	**0.626**	0.546	**0.654**
freedom	0.681	**0.738**	**0.768**	0.746	0.722	**0.742**
innov	0.764	**0.842**	**0.720**	0.590	**0.741**	0.694
honor	0.395	**0.889**	**0.553**	0.094	**0.461**	0.170
average	0.712	**0.741**	0.732	**0.736**	0.722	**0.738**

3) Comparison with Human Annotator

Because human values are unobservable private states rather than observable facts [24], 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 the inter-annotator agreement in Takayama et al. [21], [22] indicates, well trained and well qualified people will sometimes make different judgments about the same sentence. To see how our proposed classifier *SA* 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 Takayama et al. [21], [22] (These annotation by the second annotator is also available at the web page: http://nlp.inf.kyushu-u.ac.jp/values.html).

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

For this experiment, we trained our *SA* on the remaining 82 documents with meta-parameters: $\alpha = 0.6$, $FW = 4$, $FWW = 8$ (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 Tables 3.

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 of the average F_1 between human and our proposed *SA* is small. This means that *SA* effectiveness is indistinguishable from the human classifier. As can be seen, *SA* does substantially better in both precision and recall (and thus in F_1) than the second annotator on micro-average.

VI. Conclusion

Our method, which is based on simulated annealing, obtained comparable but slightly higher F_1 scores compared with those of the strong baseline by SVM. Further, in Fleischmann et al. [9], we demonstrated that the values-dictionary extracted by the proposed method can be qualitatively analyzed using thematic analysis, a commonly used qualitative method in the social sciences [2]. So far, we have only confirmed that our proposed method is applicable to the net neutrality corpus, however we have a plan to apply our method to another corpus that is related to different topic. Takayama et al. [22] have achieved slightly but statistically significantly better method in terms of classification effectiveness (F_1=0.7251 in 102-fold document cross-validation for the net neutrality corpus), which is adopted a probabilistic latent variables model. We plan to explore other methods to see if we can generate a better values-dictionary.

Acknowledgment

This work was supported in part by NSF IIS-0725459, Japangrant-in-aid for scientific research (B) 25280118, and DARPAcontract HR0011-12-C-0015.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” J. Machine Learning Research, Vol. 3, No. 4-5, pp. 993-1022, 2003.

[2] V. Braun and V. Clarke, “Using thematic analysis in psychology,” Qualitative Research in Psychology, Vol. 3, pp. 77-101, 2006.

[3] V. Černý, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm,” J. Optimization Theory and Applications, Vol. 45, Issue 1, pp. 41-51, 1985.

[4] A.-S. Cheng and K. R. Fleischmann, “Developing a meta-inventory of human values,” Proc. American Society for Information Science and Technology (ASIST2010), Vol. 47, No. 1, pp. 1-10, 2010.

[5] A.-S. Cheng, K. R.Fleischmann, P. Wang, E. Ishita, D. W. Oard, “The Role of Innovation and Wealth in the Net Neutrality Debate: A Content Analysis of Human Values in Congressional and FCC Hearings,” J. American Society for Information Science and Technology, Vol. 63, No. 7, pp. 1360-1373, 2012.

[6] A.-S. Cheng, “Values in the Net neutrality debate: Applying content analysis to testimonies from public hearings,” Doctoral dissertation, University of Maryland, College Park, MD, USA, 2012.

[7] J. Cowie, J. Guthrie, L. Guthrie, “Lexical disambiguation using simulated annealing, ” Proc. 14th Conf. on Computational linguistics (COLING '92), Vol. 1, pp. 359-365, 1992.

[8] K. R. Fleischmann, D. W. Oard, A.-S. Cheng, J. Boyd-Graber, T. C. Templeton, E. Ishita, J. A. Koepfler, W. A. Wallace, “Content Analysis for Values Elicitation,” Proc. ACM SIGCHI 2012 Conf. on Human Factors in Computing Systems, Workshop on Methods for Accounting for Values in Human-Centered Computing, Austin, TX, USA, 2012.

[9] K. R. Fleischmann, Y. Takayama, A.-S. Cheng, Y. Tomiura, D. W. Oard, E. Ishita, “Thematic Analysis of Words that Invoke Values in the Net Neutrality Debate,” iConference 2015, Newport Beach, CA, USA, 2015.

[10] T. L. Griffiths and M. Steyvers, “Finding Scientific Topics,” Proc. National Academy of Sciences of the United States of America, Vol. 101 (Suppl. 1), pp. 5228-5235, 2004.

[11] H.-F. Hsieh, and S. Shannon, “Three Approaches to Qualitative Content Analysis, ” Qualitative Health Research, Vol. 15, No. 9, pp. 1277-1288, 2005.

[12] E. Ishita, D. W. Oard, K. R. Fleischmann, A.-S. Cheng, and T. C. Templeton, “Investigating Multi-Label Classification for Human Values,” Proc. American Society for Information Science and Technology (ASIST2010), Vol. 47, No.1, pp. 1-4, 2010.

[13] T. Joachims, “Learning to classify text using support vector machines,” Springer Science+Business Media, New York, 2002.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” Science, Vol. 220, pp. 671-680, 1983.

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

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, “Equation of State Calculations by Fast Computing Machines, ” J. Chemical Physics, Vol. 21 No. 6, pp. 1087-1092, 1953.

[17] B. Pang, and L. Lee, “Opinion Mining and Sentiment Analysis,” Foundations and Trends in Information Retrieval, Vol. 2, No. 1-2, pp. 1-135, 2008.

[18] G. Sampson, “A Stochastic Approach to Parsing, ” Proc. 11th Conf. on Computational linguistics (COLING '86), pp. 151-155, 1986.

[19] S. H. Schwartz, “Value orientations: Measurement, antecedents, and consequences across nations,” In R. Jowell, C. Roberts, R. Fitzgerald, and G. Eva eds., Measuring attitudes cross-nationally: Lessons from the European Social Survey, London, England: Sage, 2007.

[20] F. Sebastiani, “Machine Learning in Automated Text Categorization,” ACM Computing Surveys, Vol. 34, No. 1, pp. 1-47, 2002.

[21] Y. Takayama, Y. Tomiura, E. Ishita, Z. Wang, D. W. Oard, K. R. Fleischmann, and A.-S. Cheng, “Improving Automatic Sentence-Level Annotation of Human Values Using Augmented Feature Vectors, ” Conf. Pacific Association for Computational Linguistics (PACLING 2013), Tokyo, Japan, Sept. 2013.

[22] Y. Takayama, Y. Tomiura, E. Ishita, D. W. Oard, K. R. Fleischmann, and A.-S. Cheng,: “A Word-Scale Probabilistic Latent Variable Model for Detecting Human Values,” Proc. 23rd ACM International Conference on Information and Knowledge Management (CIKM 2014), Shanghai, China, Nov. 2014.

[23] Y. R. Tausczik, and J. W. Pennebaker, “The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods.” J. Language and Social Psychology, Vol. 29, No. 1, pp. 24–54, 2010.

[24] J. M. Wiebe, “Tracking Point of View in Narrative,” Computational Linguistics, ” Vol. 20, No. 2, pp. 233-287, 1994.

[25] E. Woodrum, “Mainstreaming content analysis in the social science: Methodological advantages, obstacles, and solutions,” Social Science Research, Vol. 13, pp. 1–19, 1984.