# Building and Validating Hierarchical Lexicons with a Case Study on Personal Values

Steven R. Wilson, Yiting Shen, and Rada Mihalcea

University of Michigan, Ann Arbor, MI, USA
{steverw,yiting,mihalcea}@umich.edu

**Abstract.** We introduce a crowd-powered approach for the creation of a lexicon for any theme given a set of seed words that cover a variety of concepts within the theme. Terms are initially sorted by automatically clustering their embeddings and subsequently rearranged by crowd workers in order to create a tree structure. This type of organization captures hierarchical relationships between concepts and allows for a tunable level of specificity when using the lexicon to collect measurements from a piece of text. We use a lexicon expansion method to increase the overall coverage of the produced resource. Using our proposed approach, we create a hierarchical lexicon of personal values and evaluate its internal and external consistency. We release this novel resource to the community as a tool for measuring value content within text corpora.

**Keywords:** Lexicon Induction · Crowd Sourcing · Personal Values

## 1 Introduction

Content analysis of large text corpora is often a useful first step in understanding, at a high level, what people are talking or writing about. Further, it can provide a means of quantifying a person or group's focus on emotional, political, or social themes which may be of interest to researchers in the social and information sciences. While unsupervised approaches such as topic modeling [1] can be useful in discovering potentially meaningful themes within corpus, researchers often turn to lexical resources that allow for the measurement of specific, pre-defined items such as those found in the Linguistic Inquiry and Word Count [13], the General Inquirer [15], or Wordnet Domains [8]. These domain- or concept-specific tools allow for greater control over the specific type of content being measured, and the manually crafted category names provide meaningful labels for the themes being measured. Additionally, these resources are easy to use and scale to huge amounts of text.

The manual construction of these lexical resources often requires expert linguistic or domain knowledge, and so a number of semi-supervised and crowd-sourced approaches to lexicon generation have been proposed [16, 18, 7, 14, 9]. These approaches have been effective in the creation of lexical resources to measure sentiment, affect, and emotion where the categories to be measured are generally defined at the start of the process. Systems like Empath [5] allow users to quickly build new categories by providing sets of seed words that represent the desired concepts. However, it may also be useful to allow practitioners to define the set of categories to be measured later in the process for a number of reasons: the categories may not always be initially known, or researchers may decide to measure a concept at either a more general or specific granularity without creating an entirely new framework.

Rather than representing words belonging to a lexicon as a set of lists, we propose using a hierarchical tree structure in which any node can be represented by a combination of the nodes that are its descendents. This allows for explicit modeling of hierarchical relationships between concepts, and facilitates a configurable level of specificity when measuring concepts in the lexicon. For example, one researcher may want to measure positive emotions broadly, while another may want scores for more specific dimensions such as excitement, admiration, and contentment. A well-built hierarchical lexical resource can cater to either, and once formed, can be reused for different purposes depending on the research questions being asked.

In this paper, we introduce a crowd-powered approach for the creation of such a hierarchical lexicon for any theme given only a set of seed words that cover a variety of concepts within the theme. A theme could be anything from emotion to political discourse, and as an example of this approach, we create a resource that can be used to measure the expression of personal values in text.[1] Lastly, we demonstrate an evaluation framework that can be used to verify both the internal and external validity a lexical resource constructed using our method.

## 2 Methodology

First, we collect a set of seed terms that can be used to initialize the lexicon creation process. These seeds should provide good coverage of the core concepts that will end up in the final lexical resource, but various ways of expressing these concepts do not all need to be included. We embed the seed words into a vector space and cluster them hierarchically, and reorganize the initial structure using a human-powered tree sorting algorithm. Next, we automatically expand the set of concepts to increase their coverage. The resulting expanded hierarchy can be used to measure content within texts at a configurable level of specificity.

### 2.1 Hierarchy Initialization

Before beginning the crowd-powered sorting of the concepts, we create an initial hierarchy that represents a noisy sorting the seed terms. This will greatly reduce the workload of the crowd, lowering the lexicon construction time and cost, by only tasking workers with correcting this noise rather than sorting the concepts from scratch. To create this initial hierarchical structure, we first embed each

[1] This new values lexicon, along with code that can be used to build an initial hierarchy, manage the human-powered sorting, and expand the sorted hierarchy can be found at: http://nlp.eecs.umich.edu/downloads.html

of the words or phrases from the seed set into a vector space using the Paragram model [17], which has been shown to perform competitively on a number of word- and phrase-level semantic similarity tasks. We represent phrases by averaging the vector representations of the individual words in each phrase. After obtaining the embeddings, we compute the distance between every pair of words and phrases using cosine distance, providing us with a distance matrix. Given these distances, we use the scikit-learn library [12] to perform hierarchical agglomerate clustering on the word and phrase vectors in order to generate an initial hierarchy in the form of a tree, where the leaves of the tree are the seed words and phrases. However, this organization still has room for improvement: the embedding model only loosely approximate the meanings of the seed terms and the clustering algorithm is just one step toward achieving the desired organization of the concepts. Further, the tree is binary at this stage, which may not be a flexible enough representation to capture the relationships between the seed terms.

### 2.2 Crowd Powered Concept Sorting

Next, we turn to a human powered algorithm (Algorithm 1) to improve the initial sorting. Given an algorithmically pre-sorted, unordered tree $\mathcal{T}$, we want to find a *sorted* tree $\mathcal{T}'$ such that each branch follows an organization that would be selected by a majority of human annotators. We define a *direct subtree* of a tree, $\mathcal{T}$, as a subtree, $\mathcal{S}$, of $\mathcal{T}$ such that the root of $\mathcal{S}$ is a direct child of the root of $\mathcal{T}$. We employ a recursive traversal of the tree during which each direct subtree, $\mathcal{S}$, of the current tree is sorted before sorting the current tree itself. While sorting the current tree, it is possible that new subtrees are created, which are not guaranteed to be *sorted* themselves. Therefore, we must also traverse the set of subtrees, $\mathcal{U}$, that did not originally exist in the unsorted tree $\mathcal{T}$, and sort them (or verify that they are already *sorted*).

In order to actually sort a particular tree or subtree, we first identify the current set of *groups*, $G$, which are derived from the set of *direct subtrees* of the current tree's root. Each *group* consists of one or more *group-items*, which are in turn represented as one or more *terms*. For a given *group*, the *group-items* are comprised of the set of *terms* belonging to the leaf nodes of each *direct subtree* of the *group's* root node. For example, in Figure 1, the *groups* in $G$ would be represented by subtrees with roots (1) and (2). The first *group* would consist of the *group-items* in node (1)'s direct subtrees, so the two items would be "parents" and "mother, mom, father". Regardless of the depth of a *direct subtree*, all words are combined into a single, flat list to abstract away the details of the subtree, making the sorting task less complicated for the annotators. Similarly, the second group would contain two items: "brother" and "sister".

To sort the *groups* in $G$, a Human Intelligence Task (HIT) is created in the AMT marketplace where it can be completed by crowd workers. In the sorting interface, (Figure 2) each *group* is represented as a column of stacked *group-items*, followed by an empty space where new *group-items* can be placed. Crowd workers are asked to drag and drop the *group-items* (displayed as blue boxes)

**Algorithm 1:** Crowd-powered Tree Sorting.

```
Data: $\mathcal{T}$: Tree to be sorted, $n$: number of annotators, $m$: maximum HIT
      extensions
Result: $\mathcal{T}'$: Sorted Tree
Function traverseAndSortTree($\mathcal{T}$, $n$, $m$)
    if numChildren $(\mathcal{T}) > 0$ then
        foreach $\mathcal{S} \in$ DirectSubtrees $(\mathcal{T})$ do
            $S \leftarrow$ traverseAndSortTree($\mathcal{S}$, $n$, $m$));
        $\mathcal{T}' \leftarrow$ sortSubtree ($\mathcal{T}$, $n$, $m$);
        foreach $\mathcal{U} \in$ (DirectSubtrees $(\mathcal{T}')$ $\setminus$ DirectSubtrees $(\mathcal{T})$) do
            $\mathcal{U} \leftarrow$ traverseAndSortTree ($\mathcal{U}$, $n$, $m$);
    else
        $\mathcal{T}' \leftarrow \mathcal{T}$;
    return $\mathcal{T}'$;
Function sortSubtree ($\mathcal{T}$, $n$, $m$)
    $G \leftarrow$ makeGroups (DirectSubtrees ($\mathcal{T}$));
    $H \leftarrow$ createHIT ($G$);
    $n' \leftarrow n$;
    $s \leftarrow 0$;
    while $!s$ do
        $R \leftarrow$ checkHITResults ($H$);
        if $|R| \geq n'$ then
            if majorityAgree $(R)$ or $n' \geq (m+1) \times n$ then
                $s \leftarrow 1$;
                $\mathcal{T}' \leftarrow$ mostCommon ($R$);
            else
                $H \leftarrow$ extendHIT ($H$, $n$);
                $n' \leftarrow n' + n$;
    return $\mathcal{T}'$;
$\mathcal{T}' \leftarrow$ traverseAndSortTree($\mathcal{T}$, $n$, $m$);
```

into to the configuration that they believe best represents a logical sorting of the *group-items* as semantic concepts. Within the cell representing each *group-item*, a list of up to ten randomly sampled *terms* that belong to the *group-item* are displayed so that the workers are able to glean the general concept that the *group-item* represents. Users are able to create new, empty *groups* with the click of a button, if desired. Because only one possible tree can be attained when sorting two leaf nodes (i.e., a single branch for each node), subtrees consisting of two (or fewer) leaf nodes are considered to be sorted *a priori* and do not require any human intervention.

After sorting, the users are asked to provide a label for each *group*, which can then be used as a label for the root node of the corresponding subtree. The label for a *group* could be identical to one of the *terms* belonging to the *group*

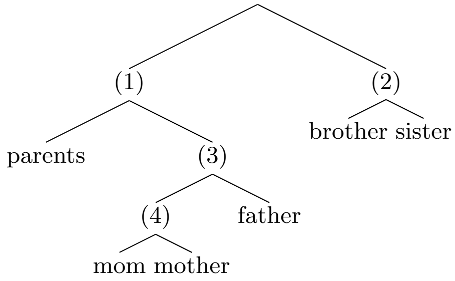
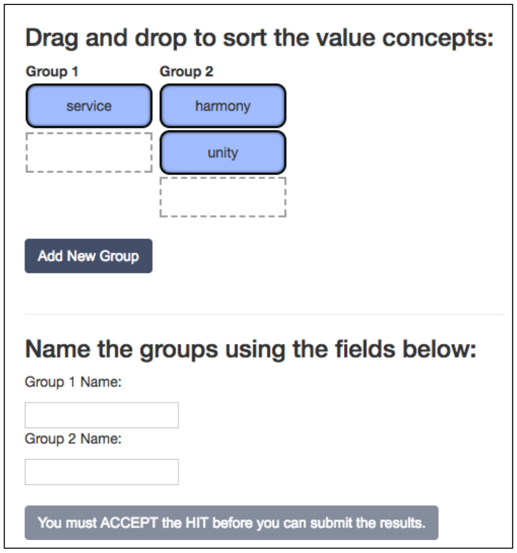


Fig. 1: Example semantic tree structure.

if the workers feel that this *term* is particularly representative of the *group*. If a *group* only contains a single *group-item* which only contains a single *term*, that *term* will remain the label for the *group* instead of adopting the crowd assigned label.

It is likely that multiple, reasonable configurations are possible. Our goal is to find the organization that is preferred by a majority of annotators. At first, we create a fixed number ($n$) of identical tasks that are required to be completed by different crowd workers. If more than $n/2$ workers sort the $group-items$ in the same way, this configuration is accepted as the majority view. However, if there is no majority view, we extend the HIT by creating $n$ additional tasks that must be completed by a new set of workers, and then checking for a majority view once again. This will be repeated a maximum of $m$ times. After $m$ HIT extensions, when all $n + n \times m$ tasks have been completed, the most common configuration is accepted as the consensus view, regardless of whether or not a majority of the workers produced this result (this is done to avoid extending ambiguous HITs indefinitely). Then, from the set of results that match the consensus configuration, the most common label for each *group* is used to name the node that is the root of that *group*. All ties are broken randomly, and empty groups are ignored. When checking for consensus, the *group* labels, the order of the *groups* themselves, and the order of the *group-items* with the columns are not considered; only the unique sets of *group-items* that were assigned to each *group*. In order to encourage workers to select a reasonable arrangement of the concepts, we also advertise and provide a bonus reward for all workers who submit the configuration that eventually is chosen as the consensus.

We then translate the consensus group configuration, $G'$, into the tree by rearranging the *direct subtrees* of the tree currently being sorted to reflect the set of *groups* selected by the crowd. Recall that each *group-item* corresponds to an entire subtree in $\mathcal{T}$. A tree representing each *group* is formed by making a link between the *group* tree's root and the root of each *group-item* tree. So, the branching factor will equal the number of *group-items* that were placed into the *group*. Similarly, the current tree's root will be connected to the root of each *group's* tree, with a branching factor of $|G'|$, the number of *groups* in the



Drag and drop to sort the value concepts:

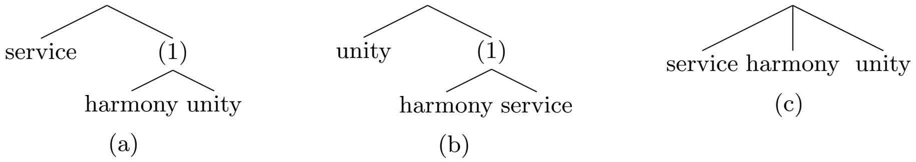
Name the groups using the fields below:

You must ACCEPT the HIT before you can submit the results.

Fig. 2: Example sorting interface

consensus configuration. Non-leaf nodes with a branching factor of one will be replaced with their children.

As an example, consider the HIT displayed in Figure 2. Figure 3 shows the trees that would result from various user actions during the sorting task. It is possible that the concepts are already sorted in a desirable configuration. Workers are not forced to make changes and are allowed to simply “verify” that the current organization is suitable (they are still asked to provide labels for the groups). The tree that would result from taking no sorting action on the example HIT is displayed in Figure 3a. On the other hand, a worker might decide that the concepts of “harmony” and “unity” do not belong together, and that “service” and “harmony” actually belong in the same grouping, separate from ”unity”. In this case, the worker can drag the box containing “harmony” into the empty cell below “service” so that these items are now members of the same *group*, resulting in tree displayed in Figure 3b. Yet another option would be to place all three items in the same *group*, which gives tree shown in Figure 3c. Note that



(a) (b) (c)

Fig. 3: Several possible tree configurations achieved by completing the same HIT in different ways.

this is equivalent to placing each *group-item* into a separate *group* of size one, since nodes with a branching factor of one will be replaced with their children. In the first two cases, the dummy label (1) in Figure 3 would be replaced with the most common text-based label assigned to the subtree by crowd workers.

### 2.3 Lexicon Expansion

Next we seek to improve the coverage of this hierarchy by expanding the set of seeds that represent a given subtree to include other semantically related words. We achieve this goal using an iterative expansion process that leverages the structure of the sorted tree. First, we obtain a vector representation for node of the tree by averaging together the embeddings of all terms contained in leaf nodes that are descendents of that node. Then, a set of candidate terms is generated by searching a set of vectors learned from a very large background corpus. A good background corpus should include examples of the seed terms in contexts that exemplify the word senses and domain in which the lexicon is intended to be applied. For example, to successfully expand a lexicon of biological terms, a background corpus of scientific literature would be more appropriate than a news corpus. For a given node vector, the top $k$ most similar word vectors to the node vector are selected as the expansion candidates (the node's expansion list).

If all candidates were accepted with a large enough $k$, it is very likely that siblings, or even distant nodes in the hierarchy, would shave intersecting sets of expanded terms. We would like to avoid accepting candidates that already belong to a sibling or another distant node, as this will lead to blurred boundaries across branches, and each node may no longer express a distinct, semantically coherent concept. This situation could be avoided by choosing a sufficiently small $k$, but this would also decrease the coverage of the lexicon. To remedy this, we examine each expansion candidate, one at a time, and determine which nodes it should belong to.

Iterating through the expansion candidates for a given node in order of their cosine similarity to the node vector (most similar first), we check if the current candidate is also a candidate for any other nodes. If it is not, then we accept the candidate as a new member of the list of words that can be used to represent the node. If all other nodes with the candidate in their expansion lists are either ancestors or descendents of the current node, we will also accept the node since it is reasonable that either more general or specific concepts will have some overlap with one another (e.g., a category about *animals* and a category about *mammals* might both contain the words "whale" and "cat", although the *mammals* category should not include "chameleon" even if this is a good word for the *animals* category). Otherwise, we only accept the candidate if it is closer to the current node than it is to any other node. If it is not, we say that the expansion for the current node has "collided" with that of another node, and we stop considering candidates for this node. The final set of words used to represent any node in the hierarchy then becomes the union of all expanded terms that belong the the subtree of which the target node is the root. For an even cleaner final sets of words, human annotators can be tasked with manually removing noisy terms, as is done by the Empath system [5]. However, the authors of that work show that this filtering has a very small effect on the final scores procured when measuring the lexical categories in text.

### 2.4 Using a Hierarchical Lexicon

A category can be selected by choosing a target node that represents the category, and a score can be assigned to any piece of text for any category by computing the frequency of words and phrases in the text that belong to the category. As before, words the belong to a category are found by taking the union of all terms in leaf nodes that are descendents of the category's root node. To increase coverage even further (at the loss of syntactic form), words in both the lexicon and the target text can be lemmatized before frequencies are calculated. Due to the hierarchical structure of the lexicon, scores for more general or more specific versions of any category can be quickly obtained by selecting a higher or lower node in the hierarchy.

## 3 Evaluating Lexicons

We explore a series of evaluation methods to test the effectiveness of any newly created hierarchical lexicon. Each of these evaluations can be generally applied to any dictionary-like lexical resource. With these methods, we seek to answer the following three evaluation questions:

1. *Does the lexicon produce reasonable scores for documents that are known beforehand to be related to the theme of the lexicon?*
2. *Are the categories in the lexicon comprised of semantically coherent sets of words?*
3. *Do the categories in the lexicon actually measure meaningful concepts?*

A good hierarchical lexicon should lead to an answer of "yes" to each question. In the following sections, we describe approaches that can be used to quantitatively answer them.

### 3.1 Frequency Testing

As a simple yet informative first step, we measure the frequency of a set of pre-selected categories on documents that are known to be related to concepts in the lexicon. This will provide a preliminary understanding of the coverage and relative scores produced by the new resource, and it will help us to answer the first evaluation question. For example, a lexicon created to measure political language should certainly produce non-zero scores for many categories when applied to a corpus of political texts. Further, documents from left-wing media sources should achieve higher scores for categories intended to measure concepts such as liberalism than categories about conservative politics.

### 3.2 Word Intrusion Choose Two

Next, we employ a coherence method borrowed from the topic modeling literature: Word Intrusion Choose Two (WICT) [10], which is a modified version of the Word Intrusion task [3]. The premise of this approach is that for a set of semantically related words, it should be easy for humans to detect randomly inserted words that do not belong to the set. Coherence is determined by presenting some words from the same category to human judges along with an *intruder* word that does not belong to that category. The *intruder* should be a word that is semantically distant from the category being evaluated, but it should be a member of one of the other categories (otherwise, the *intruder* might be easy to detect simply because it is not related to the theme or the lexicon at all, or it may be a very uncommon word). If most, or all, of the human judges can correctly identify the *intruder*, then the set of true category words is said to be "coherent". This coherence is quantified for category $c$ within model $m$ using the Model Precision measure:

$$MP_c^m = p_{turk}(\mathbf{w}_{c,i}^m)$$

where $\mathbf{w}_c^m$ is the set of words chosen to represent category $c$ by model $m$, $p_{turk}(\mathbf{w}_{c,k}^m)$ is the observed probability of a crowd worker selecting the $k$th word in $\mathbf{w}_c^m$ as an intruder word, and $i$ is the index of the *intruder* word.

WICT adds a slight modification to this: for each category, judges are asked to identify *two intruders* even though only one actually exists. For a coherent category, two conditions must be met: First, all (or most) of the human judges should choose the true intruder as one of their guesses; second, the judges' other guesses should follow a uniform random distribution across all of the true category words. If any of the true category words is selected much more often than the others, then this word does not appear to semantically fit quite as well as the others. To quantify the coherence of a category, Model Precision Choose Two for category $c$ within model $m$ is computed as:

$$MPCT_c^m = H(p_{turk}(\mathbf{w}_{c,1}^m), \ldots, p_{turk}(\mathbf{w}_{c,n}^m))$$

where $H(\cdot)$ is the Shannon Entropy [4], and $n$ is the total number of words displayed to the judges. Higher values indicate more even distributions, and therefore more coherent categories.

Concretely, each time that we test a category's coherence, we select five words from that category and an intruder word from another category (that is not also a member of the category being tested). These words are then presented to ten human judges on the AMT platform, and each judge is asked to label two intruders. As an attention check, we also randomly insert sets in which four highly related words are presented with two very unrelated words. We do not use scores provided by judges who fail these attention checks. Finally, we compute $MPCT_c^m$ for a set of pre-selected categories from the hierarchical lexicon in order to answer our second evaluation question.

### 3.3 Category-Text Matching

Lastly, we aim to answer the third evaluation question by determining how well the categories of our new lexicon actually capture meaningful concepts. To quantify this, we first select a set of interesting categories from the lexicon. Next, we obtain scores for each of these categories across text corpus in order to find the documents that have high, middle, and low scores for each category. To test a category, we select two documents: one that has a high score for that category and another than doesn't. These two documents are presented to a set of judges on AMT who are given the category label and asked to decide which document best expresses the concept described by the label. If the judges can select the correct document significantly more than half of the time, we know that the lexicon is able to identify text that expresses the category being evaluated. There are two settings for Category-Text Matching: *high-low* and *high-median*. In *high-low*, one of the top $q$ scoring documents is paired with one of the bottom scoring $q$ documents for the category, while high-median pairs this same high-scoring document with one of the $q$ documents surrounding the median scoring document. The score for either version of the task is reported as the percentage of judges who correctly selected the high-scoring text. In each HIT, a crowd worker is shown seven pairs of texts, one of which is a randomly inserted checkpoint question based on a Wikipedia article title and contents: the title of the article is shown, and the first paragraph of the article is shown as one choice while the first paragraph of a *different* article is shown as an alternative. HIT are rejected when workers are unable to identify the correct article.

## 4 Case Study: A Lexicon for Values

Previous lexical resources have been created to measure moral values [6] and tools like the Linguistic Inquiry and Word Count [13] do measure some concepts that might be considered personal values, such as “family” and “work”. However, no word-level lexical resource has previously been released that focuses on a wide range of personal values. Therefore, we consider personal values as the theme for our case study, exemplifying the hierarchical lexicon creation process. In this section, we describe the process of creating and evaluating this novel resource.

### 4.1 Collecting Seed Data

In order to collect sets of English words that are known to be related to values across multiple cultural groups, we turn to four sources:

**Mobile Phone Surveys:** Using the mSurvey platform, we distributed short surveys to 500 participants each in Kenya, the Phillipines, and Trinidad and Tobago. Respondents were paid a fee via their mobile phone to respond with text messages listing the values that are most important to them. Each respondent provided three values for a total of 1,500 value words or phrases. The phrases were manually examined and corrected for spelling mistakes. Examples of values collected include: *peace*, *harmony*, *patience*, *family*, and *money*.

**Online Value Surveys:** We use the text data from [2] in which participants recruited via Amazon Mechanical Turk (AMT) were asked to write about their personal values for 6 minutes. Respondents were from both the United States and India. We extract all unigrams and bigrams that appear at least 10 times in this corpus and add them to our set of seed words. Some of the seed words and phrases extracted from this data set are: *children*, *wisdom*, *nature*, *honesty*, and *dignity*.

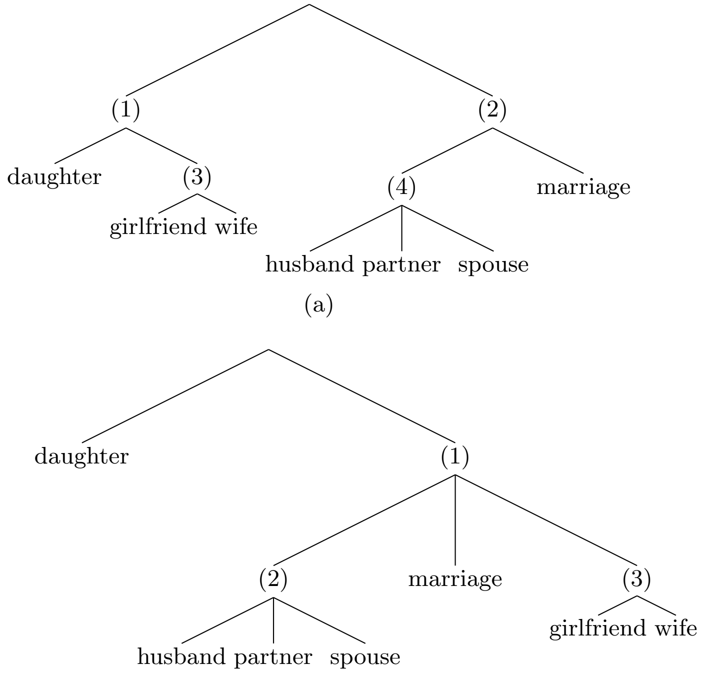
**Abridged Value Surveys:** We also collected additional surveys from the United States and India in which AMT workers were asked to list their three most important values. We collected 500 such surveys from each country, for a total of 3,000 additional value words and phrases. Here, the respondents shared that things such as *hard work*, *love*, *kindness*, *belief in god*, and *integrity* were important to them.

**Templeton Foundation Values:** Sir John Templeton formulated a list of 50 terms thought to outline values that people hold. We add this list of terms to our seed set, as well. Some examples of these items are *optimism*, *spirituality*, *generosity*, *courage*, and *creativity*.

In the end, we remove duplicate value words and phrases and manually correct the items for spelling and grammatical errors. At the end of this process, we are left with 376 value words and phrases due to a high number of duplicate answers. Collecting these responses from a range of diverse populations means that the set of words represent concepts that are important to people in many cultures.

### 4.2 Organizing the Value Words

When sorting the concepts in the values hierarchy, we initially collect $n = 5$ results per HIT for a maximum of $m = 10$ results per HIT. The average proportion of workers that selected the consensus configuration was 0.530, and the consensus configuration was chosen as the result of breaking a tie with a frequency of 0.11. Many cases requiring a tie-breaker are somewhat ambiguous, such as the two alternatives depicted in Figure 4 (an actual example of a tie that had to be broken while creating the values lexicon; each configuration was submitted by three workers). One configuration (Figure 4a) appears to group the words by gender, while the other (Figure 4b) groups the words by the type of relationship: romantic partner and child. Due to a high amount of noise in the mturk workers' node labels, we manually corrected or replaced a number of them to get cleaner category names. After viewing the hierarchy, we also manually moved a small number of subtrees to account for long-distance relationships that the mturk workers were not able to consider because of their narrow view of the overall tree structure. For the lexicon expansion, we find the counter-fitted paragram vector space [11] provided the cleanest and most coherent sets of expansion candidates. We set the number of expansion candidates at $k = 100$.



(a)

(b)

Fig. 4: Two equally common configurations submitted for the same set of nodes.

### 4.3 Evaluation

For the Frequency Testing evaluation, we collect a corpus of recent posts from a set of Reddit[2] online communities (subreddits) focused on topics that are expected to be related to personal values (e.g., /r/family, /r/christian) and apply the lexicon to these texts in order to verify that categories related to the community are expressed to a higher degree than other categories (Table 1). Many of the results are expected, such as high scores for the Religion category (includes words like *pray*, *jesus*, *divinity*) in the /r/christian category and high scores for the Wealth category (includes *revenue*, *wage*, and *cash*) in the /r/money posts. Interestingly, the Relationships category, which is a supercategory of the Family category, actually has the highest score for the posts in /r/family. This is likely because the Relationships category contains words from the Family category in addition to others like *companion*, *buddy*, and *coworker*.

For the Word Intrusion Choose Two task, we evaluate each category five times, each time querying ten unique judges on AMT. The scores in Table 2 show the regular Model Precision (MP; frequency with which judges correctly identified the intruder) and the entropy-based Model Precision Choose Two (MPCT) score described in Section 3.2. The baseline for MP is random guessing, and for

[2] reddit.com

| | Cognition | Emotion | Family | Learning | Optimism | Relationships | Religion | Respect | Society | Wealth |
|---|---|---|---|---|---|---|---|---|---|---|
| /r/christian | 1.96 | 0.68 | 0.92 | 0.56 | 0.19 | 1.82 | **6.26** | 1.51 | 3.74 | 0.48 |
| /r/college | 1.34 | 0.57 | 0.39 | **3.73** | 0.10 | 0.95 | 0.26 | 1.79 | 3.08 | 1.26 |
| /r/finance | 1.29 | 0.29 | 0.09 | 1.26 | 0.17 | 0.58 | 0.04 | 1.01 | 2.07 | **3.20** |
| /r/family | 1.54 | 0.60 | 5.58 | 0.60 | 0.10 | **7.20** | 0.10 | 2.04 | 3.55 | 0.89 |
| /r/love | 2.63 | 1.21 | 0.39 | 0.33 | 0.23 | 1.79 | 0.85 | 1.75 | **4.72** | 0.39 |
| /r/mentalhealth | 2.43 | 1.20 | 0.57 | 0.40 | 0.18 | 1.12 | 0.05 | 1.62 | **3.77** | 0.73 |
| /r/mom | 1.36 | 0.50 | 4.38 | 0.51 | 0.10 | **5.08** | 0.08 | 1.73 | 3.93 | 0.91 |
| /r/money | 1.58 | 0.16 | 0.42 | 0.61 | 0.06 | 0.91 | 0.00 | 1.13 | 2.94 | **5.29** |
| /r/parenting | 1.23 | 0.38 | 3.92 | 0.68 | 0.12 | **5.08** | 0.10 | 1.78 | 2.76 | 0.81 |
| /r/positivity | 2.35 | 1.05 | 0.36 | 0.46 | 2.74 | 1.13 | 0.48 | 1.40 | **4.71** | 0.64 |
| /r/work | 1.25 | 0.38 | 0.21 | 0.44 | 0.10 | 0.73 | 0.03 | 1.75 | **2.98** | 1.22 |

Table 1: Average category word frequency × 100 for selected value categories measured on content from various topical online communities.

MPCT it is the lower bound achieved by repeatedly selecting the same term, causing the greatest imbalance in the distribution. Art and Family are some of the most semantically coherent categories, while Respect is the least coherent.

Finally, we evaluate using Category-Text Matching in both the *high-low* (CTMhl) and *high-median* (CTMhm) settings. For this, we use the same Reddit corpus as the Frequency Testing evaluation and set $q = 5$ (i.e., we select one of the top 5 scoring texts for the category and compare it with one of the middle/bottom 5 scoring texts). We evaluate the same set of categories as were used in the WICT experiments. We evaluate each category five times, using ten judges each time. The scores reported in Table 2 are the per-category averaged scores across all judges and trials. For both settings, the baseline is random guessing. The high-scoring Religion and Siblings texts were easiest for human judges to differentiate from other texts, while high scoring Work-ethic and Order texts were essentially indistinguishable from random texts, indicating that these categories are unreliable and may need to be removed from the final set of categories to be used.

## 5 Conclusions

We have proposed a methodology for the creation of hierarchical lexicons with any theme, including a crowd-powered sorting algorithm and tree-based lexicon expansion. Researchers only need to provide a set of seed terms that are related to the theme of the lexicon and provide some high-level oversight during the lexicon creation process. To show the utility of this approach, we create a lexical resource for the measurement of personal values in text data and release this resource to

| **Category** | **MP** | **MPCT** | **CTMhl** | **CTMhm** | **Category** | **MP** | **MPCT** | **CTMhl** | **CTMhm** |
|---|---|---|---|---|---|---|---|---|---|
| Accepting-others | 0.68 | 1.40 | 0.74 | 0.43 | Achievement | 0.82 | 1.16 | 0.93 | 0.75 |
| Advice | 0.72 | 1.16 | 0.63 | 0.44 | Animals | 0.96 | 0.59 | 0.86 | 0.93 |
| Art | 1.00 | 0.92 | 0.83 | 0.50 | Autonomy | 0.80 | 0.80 | 0.50 | 0.83 |
| Career | 0.90 | 1.13 | 1.00 | 0.96 | Children | 0.94 | 1.14 | 0.91 | 1.00 |
| Cognition | 0.94 | 1.32 | 0.76 | 0.44 | Creativity | 0.84 | 1.02 | 0.64 | 0.73 |
| Dedication | 0.92 | 1.39 | 0.85 | 0.50 | Emotion | 0.82 | 1.29 | 0.68 | 0.46 |
| Family | 0.95 | 0.87 | 0.85 | 1.00 | Feeling-good | 0.92 | 1.01 | 0.70 | 0.69 |
| Forgiving | 0.90 | 1.02 | 0.64 | 0.95 | Friends | 0.74 | 0.92 | 0.65 | 0.72 |
| Future | 0.62 | 1.29 | 0.58 | 0.65 | Gratitude | 0.94 | 0.93 | 0.42 | 0.64 |
| Hard-work | 0.90 | 1.01 | 0.71 | 0.52 | Health | 0.96 | 0.43 | 0.71 | 0.95 |
| Helping-others | 0.86 | 1.37 | 0.36 | 0.31 | Honesty | 0.94 | 1.07 | 0.67 | 0.78 |
| Inner-peace | 0.70 | 1.01 | 0.96 | 0.24 | Justice | 0.82 | 1.29 | 0.43 | 0.39 |
| Learning | 0.84 | 0.86 | 0.97 | 0.61 | Life | 0.74 | 1.27 | 0.89 | 0.26 |
| Marriage | 0.80 | 0.90 | 0.93 | 0.69 | Moral | 0.92 | 1.19 | 0.54 | 0.67 |
| Optimism | 0.84 | 0.93 | 0.96 | 0.91 | Order | 0.90 | 1.05 | 0.54 | 0.30 |
| Parents | 0.80 | 0.99 | 0.77 | 0.91 | Perseverance | 0.94 | 1.04 | 0.68 | 0.23 |
| Purpose | 0.64 | 0.83 | 0.38 | 0.30 | Relationships | 0.92 | 1.06 | 1.00 | 0.78 |
| Religion | 0.66 | 1.26 | 1.00 | 1.00 | Respect | 0.36 | 1.03 | 0.11 | 0.48 |
| Responsible | 0.60 | 1.06 | 0.77 | 0.65 | Security | 0.78 | 1.11 | 0.83 | 0.64 |
| Self-confidence | 0.78 | 0.91 | 0.85 | 0.75 | Siblings | 0.68 | 0.91 | 1.00 | 1.00 |
| Significant-others | 0.89 | 0.81 | 0.71 | 0.73 | Social | 0.63 | 1.11 | 0.84 | 0.75 |
| Society | 0.68 | 0.69 | 0.07 | 0.54 | Spirituality | 0.68 | 0.85 | 0.65 | 0.83 |
| Thinking | 0.90 | 1.37 | 1.00 | 0.92 | Truth | 0.68 | 1.11 | 0.63 | 0.81 |
| Wealth | 0.96 | 0.69 | 1.00 | 0.92 | Work-ethic | 0.86 | 1.15 | 0.45 | 0.50 |
| | | | | | *Baseline* | *0.33* | *0.00* | *0.50* | *0.50* |
| | | | | | **Average** | **0.81** | **1.04** | **0.66** | **0.72** |

Table 2: Word Intrusion and Category-Text Matching results for each value category.

the community. The values lexicon achieves promising results across a series of evaluation methods designed to test both intrinsic and extrinsic validity.

## Acknowledgements

This material is based in part upon work supported by the Michigan Institute for Data Science, by the National Science Foundation (grant #1344257), and by the John Templeton Foundation (grant #48503). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Michigan Institute for Data Science, the National Science Foundation, or the John Templeton Foundation.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
2. Boyd, R.L., Wilson, S.R., Pennebaker, J.W., Kosinski, M., Stillwell, D.J., Mihalcea, R.: Values in words: Using language to evaluate and understand personal values. In: ICWSM. pp. 31–40 (2015)
3. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Advances in neural information processing systems. pp. 288–296 (2009)
4. Cover, T.M., Thomas, J.A.: Elements of information theory. John Wiley & Sons (2012)
5. Fast, E., Chen, B., Bernstein, M.S.: Empath: Understanding topic signals in large-scale text. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 4647–4657. ACM (2016)
6. Graham, J., Haidt, J., Nosek, B.A.: Liberals and conservatives rely on different sets of moral foundations. Journal of personality and social psychology **96**(5), 1029 (2009)
7. Igo, S.P., Riloff, E.: Corpus-based semantic lexicon induction with web-based corroboration. In: Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics. pp. 18–26. Association for Computational Linguistics (2009)
8. Magnini, B., Cavaglia, G.: Integrating subject field codes into wordnet. In: LREC. pp. 1413–1418 (2000)
9. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word–emotion association lexicon. Computational Intelligence **29**(3), 436–465 (2013)
10. Morstatter, F., Liu, H.: A novel measure for coherence in statistical topic models. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 543–548 (2016)
11. Mrkšić, N., Séaghdha, D.O., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.H., Vandyke, D., Wen, T.H., Young, S.: Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892 (2016)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of liwc2015. Tech. rep. (2015)
14. Rao, D., Ravichandran, D.: Semi-supervised polarity lexicon induction. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. pp. 675–682. Association for Computational Linguistics (2009)
15. Stone, P.J., Bales, R.F., Namenwirth, J.Z., Ogilvie, D.M.: The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. Systems Research and Behavioral Science **7**(4), 484–498 (1962)
16. Thelen, M., Riloff, E.: A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 214–221. Association for Computational Linguistics (2002)

17. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. arXiv preprint arXiv:1511.08198 (2015)
18. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. pp. 347–354. Association for Computational Linguistics (2005)