PerspectiveSpace

by

Jason Bernardino Alonso

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author ....
Program in Media Arts and Sciences, School of Architecture and Planning
August 28, 2008

Certified by ....
Henry A. Lieberman
Research Scientist
Thesis Supervisor

Accepted by ....
Deb K. Roy
Chair, Academic Program in Media Arts and Sciences

PerspectiveSpace

by

Jason Bernardino Alonso

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on August 28, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Words mean different things to different people, and capturing these differences is often a subtle art. These differences are often "a matter of perspective," and perspective can be taken to be the set of beliefs held by a person as a result of their background, culture, tastes, and experience.

Understanding perspective is often pivotal in resolving disputes, and such an understanding is useful for estimating the opinions a person would have on a matter, making recommendations to the person, and even understanding the language the person uses. Traditionally, perspectives are studied through the use of questionnaires and surveys that rarely leave people feeling like their opinions have been properly represented.

In this paper, I propose a system for discovering distinct communities of people with coherent belief patterns, while providing a means to characterize those patterns. This system utilizes data on how people agree or disagree on assertions that they themselves have expressed. This system, called PerspectiveSpace, is an approach whereby elementary linear operations are used to perform calculations on user models and microtheories.

PerspectiveSpace has applications ranging from discovering subcultures in a larger society to building community-driven web sites that adapt to individual perspectives, and three such applications are illustrated here. The first is the detection and amelioration of abusive user activity on a web site with community-generated content. The second is SlantExplorer, which is a tool that highlights and analyzes the perspectives underlying a document. The third is 2-wit, a novel movie recommender based on the perspectives people have about movies rather than simply the ratings they give them.

Thesis Supervisor: Henry A. Lieberman
Title: Research Scientist

PerspectiveSpace

by

Jason Bernardino Alonso

The following served as a reader for this thesis:

Thesis Reader . . . .

Deb K. Roy
Associate Professor
MIT Media Laboratory

PerspectiveSpace

by

Jason Bernardino Alonso

The following served as a reader for this thesis:

Thesis Reader . . . .

Alex (Sandy) Pentland

Toshiba Professor of Media, Arts, and Sciences

MIT Media Laboratory

Acknowledgments

I'd like to acknowledge and sincerely thank my adviser, Henry Lieberman, for being supportive of me, my work, and this thesis. My thesis readers deserve more than I can give for their patience and indulgence. Catherine Havasi has been an irreplaceable source of encouragement and proofreading, and she deserves an extra acknowledgment as the impetus behind the current OMCS family of projects. Robert Speer was a helpful, corroborating mathematician for some of the major assertions made in this paper. Kenneth Arnold and all the developers working with him, including Jesse Mueller, have done crucial work by building the elementary tools underlying the analyses in this research.

I would also like to thank the MIT Media Laboratory and the Program for Media Arts and Sciences for supporting me and this research. Finally, I thank the sponsors of the Laboratory, especially those who have found the importance of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The variations in people's beliefs and personalities lie at the heart of many common problems where people are trying to make use of information subject to opinion. People look at reviews of movies and products, but people don't always think the same way about them. People rarely, if ever, feel like surveys properly represent their opinions. An online forum of many users is often strewn with many disagreements, and users have difficulty navigating them to find the useful commentary amongst the noise. This is especially true for newcomers to a forum, where the reputations of the regular contributors are unknown to entrants. What is needed here is a tool that lets people express themselves honestly and then captures even the subtle differences between people in a meaningful way.

PerspectiveSpace is an analysis of person-to-person interactions that explores the similarities and differences in what people believe by discovering descriptive axes on which people can be arranged. These belief patterns underlie the different "perspectives" that people may have, which can be taken to be the set of beliefs held by a person as a result of his or her background, culture, tastes, and experience. In addition to studying the varying beliefs of different social or cultural groups, PerspectiveSpace has applications in recommender systems in that it utilizes knowledge about how people think about the items being recommended.

PerspectiveSpace is derived from the singular value decomposition (SVD) of people's agreement or disagreement with natural language statements of people's beliefs.

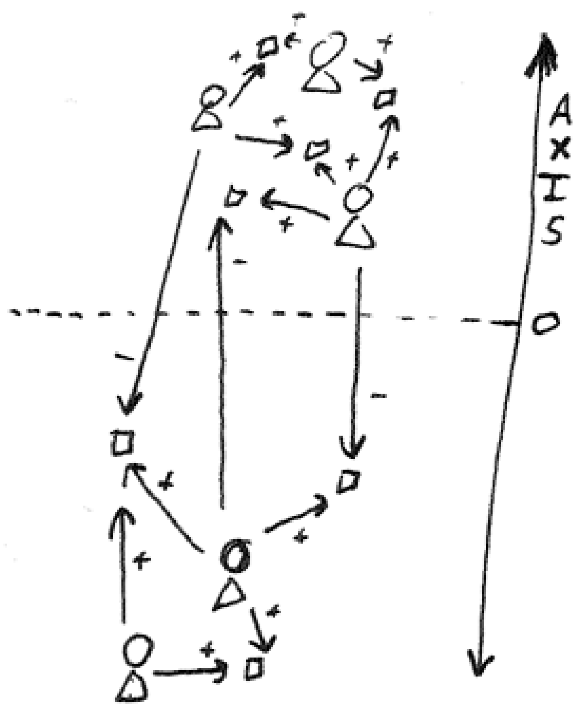


Figure 1-1: A casual representation of an axis discovered through singular value decomposition (SVD)

In particular, the subjects must be able to express their beliefs in free form text in addition to considering other the beliefs of other subjects as represented in free form text. The body of knowledge used for much of the discussion in this paper is the OpenMind Common Sense (OMCS) [25] corpus as collected through OpenMind Commons [27], as this corpus satisfies the qualifying parameters for calculating a PerspectiveSpace.

AnalogySpace is a transformation of the scored assertions (concepts crossed with features, giving scores) in ConceptNet [13] that yields a compressed vector for each concept or feature, permitting elementary linear operations to be used to perform calculations on semantic similarity [30]. PerspectiveSpace, which is separate from but related to AnalogySpace, is a transformation of the ratings (users crossed with assertions, giving ratings) that gives a compressed vector for each statement and person. The axes of these vectors represent significant variations which can be used to characterize different subcultures.

The effect of this transformation is illustrated in figure 1-1. Here, two groups of users are shown, each set agreeing with one set of assertions and collectively disagreeing with the other set of assertions. The SVD has arranged these two clusters of users and assertions along an axis, which, if the assertions within each group were considered, would likely reveal a divisive issue between the two groups of people. It should be noted that the graph of ratings is not fully connected, though the predictive power of SVD has interpolated the missing ratings by placing the people and assertions at nontrivial locations on the same axis. Not included in the illustration are the majority of assertions and users who are placed trivially on this axis, and as such these users and assertions would appear as a cloud around the origin.

There are a couple notable properties of the SVD that are important to both PerspectiveSpace and AnalogySpace. One is that the principle components found (termed "axes" in the lingo of AnalogySpace and PerspectiveSpace) are ordered in decreasing significance, and that degree of significance is measured[1]. The most significant axis divides the data into the two most divisive sets of items and their properties, summarizing groups of properties with a single varying parameter. Subsequent axes divide and describe the data along successively less significant parameters. Each of these varying parameters can be take to describe a group of properties holistically. Another important property is that the discovered axes are orthogonal, which means that each successively less significant axis describes successively more subtle variations in the data. In appropriate circumstances, the most subtle variations can be taken to be noise, which can then be removed to make sensible interpolations of missing data.

This paper begins with a technical discussion of the construction of PerspectiveSpace, followed by a survey of its implications for possible applications in general. Then, some potential applications in ethnography, recommender systems, and community-driven content systems are considered.

This work is part of the Common Sense Computing initiative, an effort of the Software Agents group to collect and apply knowledge of the way people think about their world.

[1] In an SVD, this degree of significance is called a "singular value."

Chapter 2

Related work

PerspectiveSpace is built on old and well-studied ideas in recommender systems and psychometrics, though it was motivated by (and uses the tools of) ongoing research in collecting and applying common sense knowledge. There is existing work that studies the balkanization of contributors in a collaborative content system, but not with the scale or form of PerspectiveSpace.

2.1 Common sense knowledge

The roots of the PerspectiveSpace project lie in research into collecting and applying common sense knowledge. Two leading efforts on this front are Cyc and OpenMind Common Sense (OMCS), the latter of which is directly affiliated with Perspective-Space. Though Cyc and OMCS have radically different design principles and incompatible architectures, Cyc is also discussed here, as PerspectiveSpace offers OMCS an analogue for a structure previously implemented only in Cyc and systems of a similar architecture.

2.1.1 Commonsense Computing Initiative

The Common Sense Computing initiative is an effort of the Software Agents group at the MIT Media Lab to collect and apply knowledge of the way people think about their world. A number of projects fall under the umbrella of this initiative, PerspectiveSpace included, and a few of them should be understood in the context of PerspectiveSpace.

OpenMind Common Sense

OpenMind Common Sense (OMCS) [25] is a project that seeks to collect a large body of common sense knowledge in natural language from volunteer contributors over the Internet. Launched in 2000, it grew to include over 750,000 assertions from 40,000 contributors in the English language alone. It has inspired sister projects in other languages, including *OpenMind no Brasil* [2] in Brazilian Portuguese, and the OMCS project is now working on Dutch and French incarnations. Improving the process of gathering knowledge and developing tools and applications for using it provide the major impetus for the work done in the Commonsense Computing Initiative.

OMCS is representative of what is often called "scruffy AI." This reflects the fact that OMCS does not capture knowledge in a form that readily supports manipulations using formal logic, nor does it make strong guarantees of quality in the data that it collects. In fact, the knowledge collected in the OMCS corpus is not directly machine-usable: it is in pure natural language, though projects that are a part of the Commonsense Computing Initiative provide tools and applications for the OMCS corpus that still function in the spirit of imprecision inherent in scruffy AI.

What Would They Think?

Not unrelated to the OMCS project is a system developed by Hugo Liu called "What Would They Think?" [21]. This system attempted to address the notion of perspective directly in an application where a panel of virtual "advisers" would indicate their opinions of a document under analysis in the form of visually-represented affective reactions.

Though ostensibly quite similar in purpose to SlantExplorer, which is a PerspectiveSpace application discussed in chapter 7, "What Would They Think?" (WWTT) was built around distinctly different principles, differing in their fundamental structure, scale, and output.

In terms of structure, Liu's work determined perspectives through natural language processing of representative text samples. This information was used to build an elaborate model of a "bipartite affective memory system." PerspectiveSpace, on the other hand, offloads the burden of natural language understanding to humans in the form of noting where people agree or disagree on various assertions. WWTT, furthermore, uses episodic memory as its basis for analysis, whereas PerspectiveSpace uses belief patterns in the form of the acceptance or rejection of assertions.

In terms of scale, WWTT requires representative text samples of each virtual adviser to build a model. This does not scale as readily to large populations as PerspectiveSpace's model of data collection, which is more compatible with the style of frequent, brief self-expression seen in Web 2.0 culture.

Lastly, WWTT's output differs from PerspectiveSpace and SlantExplorer by aiming to provide multiple affective measures on a document, while PerspectiveSpace and its applications operate exclusively around agreement, disagreement, or neutrality.

ConceptNet

ConceptNet [4] is a semantic network designed to be a machine-usable representation of the corpus of knowledge captured by the OMCS project. The nodes of the semantic network are normalized strings of natural language, called "concepts," and these concepts are interconnected with labeled directed links like those shown in table 2.1.

Relation type	Meaning
`MadeOf`	What is it made of?
`IsA`	What kind of thing is it?
`UsedFor`	What do you use it for?
`CapableOf`	What can it do?
`PartOf`	What is it part of?

Table 2.1: A sampling of relation types supported by ConceptNet and collected by OMCS, taken from [29]

It must be stressed that ConceptNet still does not permit formal reasoning. This network is highly linguistic in nature and lends itself most readily to language research [13, 30, 12], lending credence to the idea that language is a part of common sense.

There are two generations of ConceptNet of note: ConceptNet 2 and ConceptNet 3.

ConceptNet 2 The first public instantiation of ConceptNet was ConceptNet 2. It was built as a combination data set and toolkit with some rudimentary natural language processing abilities [22]. These features included part-of-speech tagging (with minor support from the ConceptNet data set), basic affect sensing, topic gisting, and a framework for building additional tools using spreading activation.

ConceptNet 3 The ConceptNet 2 implementation did not include a direct connection to the OMCS data set, nor did it readily permit its natural language processing components to be swapped out to support advances in natural language processing or even to support languages other than English. This author designed a new framework, the Common Sense Application Model of Architecture (CSAMOA), as a means to separate the underlying semantic network that is ConceptNet from the tools that support and use it while establishing a direct connection between ConceptNet and the OMCS data set [1]. This work led to the development of ConceptNet 3 [13] and a major revision to the OMCS collection system, which is now known as OpenMind Commons [27].

Commons

Robert Speer introduced a radical revision to the collection process utilized by OMCS. His system adapted the practice of using frames to prompt contributors[1] from generating exclusively natural language statements to generating connections in ConceptNet 3 with back references to natural language statements. Another contribution was to improve the quality of assertions contributed by showing contributors, interactively, how their contributions were improving our ability to use common sense. Built on

[1]A frame, in this context, is a natural language statement with two blanks design to extract a relationship between two concepts. For example, "A _____ is used for _____," is a typical frame.

CSAMOA and ConceptNet 3, Commons also used concurrent advances in the Commonsense Computing Initiative that eventually became known as AnalogySpace.

AnalogySpace

AnalogySpace [30, 28] is a project that uses singular value decomposition (SVD) of matrices generated from the ConceptNet data set to build a model of semantic similarity that can be manipulated with basic vector mathematics. The principles and techniques used in AnalogySpace were adapted for many similar features of PerspectiveSpace, and this project was the dominant source of inspiration for PerspectiveSpace.

Divisi

Divisi [15] is the sparse tensor and SVD toolkit developed at the MIT Media Lab for the purpose of supporting AnalogySpace, PerspectiveSpace, and other related research. It also drives the inference engine in OpenMind Commons using a method analogous to predictive ratings in PerspectiveSpace, which are discussed in section 4.2.

2.1.2 Cyc

Cyc [19] is a common sense knowledge collection project run by Cycorp, Inc. wherein factual knowledge about the world is compiled into a massive body of assertions in formal logic. Cyc is built around an ontology that organizes these assertions into modular microtheories, each of which are a small body of assertions that combine to represent a particular domain in common sense knowledge.

2.2 Psychometrics

Psychometrics is a field of psychology dedicated to psychological measurement, perhaps the most popularly-known manifestation of which is the IQ test. Early pioneers in the field of intelligence testing even developed statistical methods, namely factor analysis, that are substantially similar to principle components analysis, though important technical distinctions exist [5].

Psychometrics shares some notable features with PerspectiveSpace. In addition to the more intelligence-based measurements, the field also includes personality testing, like the Myers-Briggs Type Indicator (MBTI) [9]. The general principle of breaking down the results of a battery of questions into parts (either "factors" or "components") that describe psychological features is shared by both psychometrics and PerspectiveSpace.

There are, however, numerous and critical differences between the discipline of psychometrics and PerspectiveSpace.

2.2.1 Differences from PerspectiveSpace

Though PerspectiveSpace shares notable properties with psychometric studies, there are fundamental differences in their scope, execution, and evaluation.

In terms of scope, PerspectiveSpace is an on-line system insofar as it is designed to accumulate information about people progressively and draws conclusions depending on agreement/disagreement interactions among the people it analyzes. As such, it supports applications similar to recommender systems and AI research into common sense reasoning. Psychometric studies require extensive questioning, as evident in the ubiquitous IQ test, personality tests, and educational assessment. It is also the case that PerspectiveSpace does not intend to measure targeted psychological features as is usually done in psychometrics, but rather the semantics of dominant features are to be discovered through analysis of perspectives.

In terms of execution, psychometrics is primarily interested in tests and surveys, generally involving a very large battery of carefully-prepared questions, a substantial number of which each subject is expected to answer. PerspectiveSpace, on the other hand, operates on the tenet that subjects should consider assertions made by others and contribute their own statements for consideration by others. PerspectiveSpace operates with a much lower expectation in terms of the number of answered "questions," and these questions certainly are not subject to rigorous, experimenter-driven quality control.

Psychometric tests are generally evaluated in terms of two parameters: validity and reliability. The validity of a test is a measure of how well a test measures its targeted quantities, whereas the reliability of a test is a measure of how well repeated applications of a test to the same subject produce consistent results. PerspectiveSpace's differences on these measures can be considered separately:

Validity In comparison, notions of validity in PerspectiveSpace are vague at best. PerspectiveSpace, as implemented so far, makes no claim to measure specific properties of individuals. Furthermore, little claim is made that PerspectiveSpace measures the degree of a property of an individual to any great degree of accuracy—in fact, the very manner in which PerspectiveSpace is applied is with incomplete information, such that useful interpolations can be made.

Reliability Also, reliability can be defined for PerspectiveSpace only poorly at best, as any PerspectiveSpace implementation would evolve substantially with each iteration of a "test." It is far more natural to recognize PerspectiveSpace as an iterative measurement. It is also the case that a person's perspective is expected to change, albeit slowly, in response to education, maturation, changes in taste, etc. Measuring only immutable properties of a personality is simply not a part of this project's purpose.

2.3 Recommender systems

Recommender systems, in general practice, are tools that look at the behavior (like purchasing activity) or direct input (like movie ratings) of a user to make an informed recommendation of content, products, or other entities in which the user would likely take an interest. A popular manifestation of recommender systems is collaborative filtering, wherein recommendation is accomplished by modeling the likely behavior or response from each user using the observed input and behavior of *similar* users.

A notable landmark in the field of collaborative filtering is the Tapestry project from Xerox PARC [11] in 1992. Tapestry innovated an approach to dealing with high-volume mailing lists by letting users filter only for messages that other, *similar* users have somehow indicated as interesting. A variation on this approach was the GroupLens project, which built a distributed collaborative filtering framework for Usenet netnews [24].

Chapter 3

Implementation

This chapter covers the mathematical foundations of PerspectiveSpace as well as some details on the tools and resources used in the course of its implementation.

As an example for motivating many of the computation steps in calculating PerspectiveSpace, 2-wit is introduced here, though it is given particular attention in chapter 8. 2-wit is a recommender system that recommends reviews of products in the consumer market (in this case, movies), which is distinctly different from the traditional approach of recommending products themselves.

3.1 Mathematical foundations

The mathematical preparation of PerspectiveSpace has a few key steps. First, the underlying data set must be defined, including the collection process, with attention to the type of features intended to be found with PerspectiveSpace. This leads directly into the preparation of the source matrix, which requires some attention to the degree of trust placed in the people contributing data for analysis. This matrix must then be normalized or otherwise prepared to maximize the usefulness of the computed PerspectiveSpace, which is created from the decomposition of the normalized matrix. Treatment of the decomposition is given in chapter 4.

3.1.1 The underlying data

The collection of data for PerspectiveSpace must have four features:

1. contributors must be able to express their beliefs in succinct, natural language assertions;
2. agreement and disagreement between contributors on their assertions must be readily and frequently ascertained;
3. each contribution must be linked to the identity of the contributor; and
4. each contributor may only issue one rating per assertion, though that rating may be altered over time.

Tags

It should be noted that assertions may be lightly structured with “tags” such that the assertions can be decomposed into tags and statements. In the case of 2-wit, each statement is a review made of a particular movie. The tag is either “agree” for stating that the contributor agrees with a statement or “junk” for stating that the contributor believes the statement is obscene, spam, or otherwise generally useless in the opinion of the contributor.

Interface considerations

In the case of 2-wit, the interface (see figure 3-1) presents logged-in users with a simple box for entering one or more reviews or otherwise short commentaries on a movie. All of the reviews entered by other users are visible for consideration, and there are three icons at the top of each review for users to click to express their opinion of each review: a green thumbs-up (agree), a red thumbs-down (disagree), and a yellow flag (junk). Only one of these icons per review may be chosen by a user, and the current selection is illuminated for that user, even across multiple visits to the site. The user is permitted to change their decision by clicking on a different choice or to cancel their decision by clicking on their choice again.

This interface maps to the tags system as given in table 3.1. The system presumes that a user should agree or disagree only with reviews that the user does not consider to be junk, and that the user's opinion of agreement is irrelevant for reviews that are considered to be junk. This system was chosen for its intuitive interface while still complying with the PerspectiveSpace requirements for underlying data.

No effort was made in 2-wit to determine if any given review is a duplicate of a previously-entered one. This would constitute a hard natural language processing problem to catch the even the most subtle of variations.

Figure 3-1: A screenshot of the 2-wit collection interface

Choice	Meaning	Rating coupled with tag "agree"	Rating coupled with tag "junk"
None	No choice made	none	none
Thumbs-up	Agree with review	agree	disagree
Thumbs-down	Disagree with review	disagree	disagree
Flag	Review is junk	none	agree

Table 3.1: Semantics of 2-wit rating interface

Unless the collection interface is exceptionally designed, it will be easier to ask if someone agrees with an existing statement than it is to ask them for a new one.

This is fortunate, as PerspectiveSpace needs a significant number of people to rate the same assertions in order to place them meaningfully in a coordinate space.

It should be noted that agreement and disagreement does not have to be direct or explicit from a contributor's perspective, as it is in 2-wit. An exercise, possibly even in the form of a game, can be constructed wherein a person is asked to phrase something in his or her own words. Alternative inputs like these may come at the expense of the ease of collection. Use of natural language processing technology to determine agreement or disagreement is an acceptable mechanism once the technology is developed enough to accomplish such a task accurately with a sufficiently large number of ratings per review.

Framing considerations

As with any survey, the results are very sensitive to the language used in the instructions given to contributors. The framing, as such, must be crafted carefully to match the particular application of PerspectiveSpace that is intended.

Consider the following two pieces of common sense knowledge:

1. *Soda* is a carbonated beverage.

2. *Pop* is a carbonated beverage.

The general instructions given on the OMCS collection site are to rate assertions as "Good," "Bad," or "Fix grammar." The dominant response to these instructions from bona fide contributors would be to mark both of the above assertions as "Good." This, unfortunately, does not capture the cultural differences latent in the assertions. If the instructions were, instead, to "rate each assertion as good that you would say to be true and uses the same words that you would use to say it," contributors would likely respond differently to the assertions depending on their background.

An application may warrant multiple subtle interpretations from each contributor. For the examples above, two distinct questions may be asked of a contributor: "Is this something you consider true?" and "Is this the way you would say it?" In this situation, decomposing assertions into statements and tags would be the most direct solution, provided the collection interface can reasonably accommodate the more subtle questioning. It should be noted that the contributor does not necessarily need to see all possible decisions. 2-wit, for example, used three icons to collect information for two tags.

3.1.2 Constructing the matrix

The construction of PerspectiveSpace begins with the preparation of a ratings matrix, which is denoted as M_R in this paper. This procedure varies with the use or omission of tags in the assertion structure. The basic model, without tags, shall be discussed first.

Basic model

In the basic model, M_R has the structure of people crossed with assertions, giving ratings. The rows of this matrix correspond to the people to be analyzed, while the columns of the matrix correspond to the assertions that have been rated by the people. M_R, generally constructed as a sparse matrix is populated with real numbers with the following properties:

1. $M_R[i, j] > 0$ iff person i gave assertion j a rating of agreement;
2. $M_R[i, j] < 0$ iff person i gave assertion j a rating of disagreement; and
3. $M_R[i, j] = 0$ iff person i gave assertion j no rating or a neutral rating.

In the analysis of the OMCS data set, non-empty cells of M_R were given values of unity magnitude, but the particular data set and application of PerspectiveSpace should be considered when determining the values to be assigned. For reasons covered in section 5.3.2, it may be advisable to recognize two classes of ratings: explicit ratings a person makes of an assertion presented to him or her, and the implicit rating a person makes of an assertion when that person adds that assertion to the data set. Implicit ratings, by virtue of requiring the most human effort to make, may be given higher-magnitude values in M_R. For these reasons, 2-wit gives implicit ratings a score of ± 2.0 and explicit ratings a score of ± 1.0.

Tagged model

In the tagged model, M_R has a nested description: (people crossed with tags) crossed with assertions, giving ratings. That is, the rows of the matrix correspond to every possible pair of a person and a tag, though the matrix is still sparse as in the basic model. The columns, like in the basic model, correspond to the assertions that the people have rated. Casually notated, this sparse matrix is populated as follows:

1. $M_R[(i,k),j] > 0$ iff person i gave statement j with tag k a rating of agreement;
2. $M_R[(i,k),j] < 0$ iff person i gave statement j with tag k a rating of disagreement; and
3. $M_R[(i,k),j] = 0$ iff person i gave statement j with tag k no rating or a neutral rating.

This awkward coupling of people with tags, as opposed to a coupling of tags with statements, was chosen to maximize the interpolative power of the SVD in the final stage of the analysis. By coupling tags in this manner, an SVD can detect patterns where one group of people tend to make statements that another group tends to believe are junk (or otherwise assign a different tag). This is a more powerful analysis than the basic model can yield alone. Coupling tags with statements (that is, recomposing assertions from their parts) would result in a matrix equivalent to the basic model. Another way of phrasing this coupling is to say that the statements (columns) are being described in terms of what people think of them (rows like "Alice agrees" and "Bob thinks is junk").

People read things they don't consider true all of the time, and much of that happens in the course of reasonable debate. The "junk" tag represents a far more harsh opinion: that the statement was an irrelevant comment, profane, spam, or even simply (especially in the case of 2-wit) utterly uninsightful.

As such, the “agree” tag captures the more subtle distinctions between people’s beliefs and reveal underlying issues. The “junk” tag, on the other hand, helps protect “agree” data from opinions that don’t touch on issues at all. The OMCS PerspectiveSpace does not presently support a tagged model, and as a consequence, a large number of drone accounts polarized its top three axes with malicious intent[1]. The use of a junk tag could help protect the axes of OMCS PerspectiveSpace from such distortions in the future.

The motivations for distinguishing between implicit and explicit ratings when determining the magnitudes of these values, as discussed with the basic model, still apply with the tagged model. There is an additional consideration to be made when using the tagged model, however: many applications will benefit from weighting tags differently. In the case of 2-wit, for example, indication of a user’s agreement or disagreement (that is, a rating assigned to the ‘agree’ tag), is more descriptive of a *statement* than whether or not the person considers the statement to be junk.

Additional notation

Throughout this paper, $\mathbb{P}_A$ is used to represent the set of all assertions in PerspectiveSpace (that is, the column indices of M_R), whereas $\mathbb{P}_P$ is used to represent the set of all people (the row indices of M_R under the basic model). $\mathbb{P}_T$, when applicable, is used to represent the set of all tags (used in conjunction with $\mathbb{P}_P$ to make up the row indices of M_R under the tagged model).

3.1.3 Normalization

The matrix M_R is normalized to obtain $\hat{M}_R$. In the context of PerspectiveSpace, this process reshapes the data set going into the SVD, while altering its semantics as little as possible, so as to maximize the effectiveness of the algorithm.

There are five general normalization approaches considered in this paper:

[1] It must be noted, however, that PerspectiveSpace was originally created to protect OMCS and its derived ConceptNet from this malicious activity. PerspectiveSpace effectively “took a bullet” for the benefit of the other two projects. This incident is covered in detail in chapter 6

1. identity normalization;
2. unity magnitude of user ratings;
3. mean-shifting of user ratings to zero; and
4. complementary ratings (also called "mirroring"); and
5. combined magnitude normalization with mirrored ratings.

Identity normalization

In this model, $\hat{M}_R = M_R$. Given the OMCS data set, this was able to isolate abusive user behavior (see chapter 6) on axis 18.

On the same data set, identity normalization resulted in "echoes" of axes across multiple axes. For example, an axis would show a non-trivial placement of about three people, all in agreement near the 1.0 end of the axis (the positive extreme). The next axis (in decreasing order of magnitude in singular value), however, would show the same three people, except one of them would be placed around -0.25 while the other two were still near 1.0. The natural interpretation by subculture detection (a process discussed in section 4.1) is that the third person has a slight tendency to disagree with the other two on the assertions placed non-trivially on that axis. An examination of the data, however, reveals that the negatively-placed person simply did not rate the assertions in question.

The ultimate interpretation of this phenomenon, under the identity and magnitude normalization algorithms, was that the lack of a rating was considered significant under SVD. A missing rating is equivalent to having a rating value of 0, but the average rating provided by a person tended to be significantly greater than zero[2]. Once this observation was made, identity and magnitude normalization methods were abandoned in favor of better-tailored approaches.

[2] A non-zero average rating per person is obvious given that roughly 75% to 80% of the assertions in the OMCS corpus are considered true, as shown in studies with human judges.

Unity magnitude

As mentioned above, magnitude normalization methods do not address the formation of echoes. These methods do, however, improve the quality of the discovered axes by preventing the most significant axes from being dominated by the most populated rows and columns of the input matrix. When using identity normalization in AnalogySpace, the most significant axes simply described the concepts for which OMCS had the most data [30]. In terms of 2-wit, a user who rates 100 reviews would overpower the user who rates 10 reviews in the data set, such that the user who rated 100 reviews would likely establish an axis unto himself/herself. Unity magnitude normalization rescales the input values, so users can readily establish their positions in PerspectiveSpace without simply granting the most prolific users their own axes. This is important, as the data set requirements outlined in 3.1.1 place no constraints on the relative number of ratings obtained from each person.

An immediately apparent limitation of unity magnitude normalization is that rows of the input matrix with very little content, which would describe a person (in PerspectiveSpace) who contributed a single rating, would have just as much influence in the formation of axes as rows with a lot of content. The solution adopted in AnalogySpace, and subsequently adopted in PerspectiveSpace, was to add a constant term in each row to make the magnitude of the normalized row vary with the magnitude of the initial row.

This normalization model works as follows, with b as the "base parameter" to be added to each row to diminish insignificant rows:

$$\hat{M}_R[i,j] = \frac{M_R[i,j]}{\sqrt{b + \sum_{j'' \in \mathbb{P}_A} M_R[i,j'']^2}} \tag{3.1}$$

Mean-shifting

Mean-shifting is a simple method to address the echo phenomenon that accompanies identity normalization, and this method is commonly practiced when using SVDs. Its principle is much simpler than the equations below may imply: offset each non-zero value of each row of the matrix by a fraction of the average value of that row, such that the average value of each row in the normalized matrix is zero.

In this model, $\hat{M}_R$ was calculated as follows:

$$\hat{M}_R[i,j] = \begin{cases} 0 & \text{if} M_R[i,j] = 0 \\ M_R[i,j] - \frac{1}{|\{j' | j' \in \mathbb{P}_A, M_R[i,j'] \neq 0\}|} \sum_{j'' \in \mathbb{P}_A} M_R[i,j''] & \text{otherwise} \end{cases} \tag{3.2}$$

This equation is obfuscated by the complexities of expressing a few of its terms. $|j' \in \mathbb{P}_A, M_R[i,j'] \neq 0\}|$, for one, is an elaborate way to say "the number of nonzero values in row i."

Given the OMCS data set, this model was free of the echo effect seen in with identity and magnitude normalization, revealing cleaner axes. This model, however, places undue emphasis on negative ratings. Given how 75% to 80% of the assertions in the corpus are considered true, one naturally expects most of the ratings to be made by a person to be ratings of agreement, and this is empirically verified. This also means that very few ratings by a person, if any, are negative. Mean shifting thus increases the magnitude of the negative ratings significantly, while diminishing the magnitude of the many positive ratings. Possibly even more damaging is that people who have only rated things positively will have no non-zero values in their rows of $\hat{M}_R$.

This bias toward negative ratings may have had a significant impact in the top few axes of PerspectiveSpace over OMCS. In particular, the abusive user behavior that was found on axis 18 with identity normalization is strikingly demonstrated in the top several axes of PerspectiveSpace under mean-shifting.

Complementary ratings

Under the complementary ratings model, every assertion was matched with a complementary, opposite assertion. For every rating a person gives a normal assertion, the person is modeled to give the opposite rating to the complementary assertion. This ensures that the average rating given by a user is always zero without placing undue bias on any particular assertion. As such, the echo effect that appears as a result of using the identity or unity magnitude normalization methods disappears.

Combined magnitude normalization with mirrored ratings

The final normalization method chosen for 2-wit, and recommended for any PerspectiveSpace preparation, was a combination of mirroring and unity magnitude normalization. Mirroring is applied first to ensure that the average value in each row of the resultant matrix is zero. Applying unity magnitude normalization (with a base term of 2.0 by default for the purposes of 2-wit) does not alter the zero-mean property of the matrix.

3.1.4 Decomposition

Once a normalized matrix is obtained, a singular value decomposition (SVD) is taken, yielding U, Σ, and V^T:

$$\hat{M}_R \approx U \Sigma V^T \tag{3.3}$$

U gives the coordinates of each user in PerspectiveSpace, while V gives the coordinates of each assertion in the same space. An alternate form of this equation is used at various times in this paper:

$$\hat{M}_R[i, j] \approx \mathbf{f}_P(i) \bullet \mathbf{f}_A(j) \tag{3.4}$$

Here, $\mathbf{f}_P(i)$ is the vector yielded by $U[i]\sqrt{\Sigma}$, whereas $\mathbf{f}_A(j)$ is the vector yielded by $V[j]\sqrt{\Sigma}$. When references are made in this paper to "axes," the components of these vectors are the axes in question.

For the purposes of this paper, a group of people or assertions discovered to be placed non-trivially on a PerspectiveSpace axis shall be considered to define a *principle perspective*. That is, a group of people described in this manner shall be called the *people of a principle perspective*, while a corresponding group of assertions shall called the *beliefs of a principle perspective*. The term is chosen to emphasize the similarities and differences between these axes and the concept of an ad-hoc perspective as discussed in 4.3.

3.2 Implementation details

The development of PerspectiveSpace used a specialized software library developed, in part by this author, at the MIT Media Laboratory: Divisi [15]. Divisi was developed in the programming language Python [10] for the purpose of conducting experiments with SVDs over various named entities (like persons or assertions) instead of matrix indices. This library services research on AnalogySpace and PerspectiveSpace in addition to powering a variety of AnalogySpace-based features on the Open Mind Commons web site.

For the purposes of PerspectiveSpace, the OMCS data set was used through the ConceptNet representation, for which a data model and abstraction API was developed, called the Common Sense Applications Model of Architecture (CSAMOA) [1], which underlies the development of ConceptNet's semantic network from the natural language representation in the original OMCS corpus[3]. CSAMOA uses and exposes the data abstraction system of the Django web development framework [14].

A tool called "svdvis," which was originally developed for exploring AnalogySpace, was adapted to visualize PerspectiveSpace. This is an OpenGL-based tool written in C++ by this author and Michael Morris-Pierce. This tool was useful in navigating PerspectiveSpace for research and for generating several of the figures seen in this paper.

A notable modification to svdvis from its use in AnalogySpace was the use of a non-linear mapping between PerspectiveSpace coordinates and the visualized coordinates: the magnitude of any coordinate was replaced with its square root. This allowed many points that were otherwise densely clustered around the origin to be spread out into a visual scale comparable to more distant points.

[3]ConceptNet is now the native representation of knowledge collected for OMCS, though sentences are still recorded in natural language form as well as semantic network form.

3.2.1 Open Mind Common Sense

The OMCS corpus collects information from registered contributors using an interface dominated by the features shown in figure 3-2. The effect of forcing contributors to be registered is that all contributions can be traced to a particular user account. What is immediately apparent in this interface is the ease in which a person can contribute a rating by choosing from one of three ratings, or no rating at all: agree (thumbs up), disagree (thumbs down), and fix grammar (flag)[4]. The bottom portion of the interface shows boxes containing predicted knowledge (a feature of AnalogySpace) that can be edited to contain an arbitrary assertion for submission.

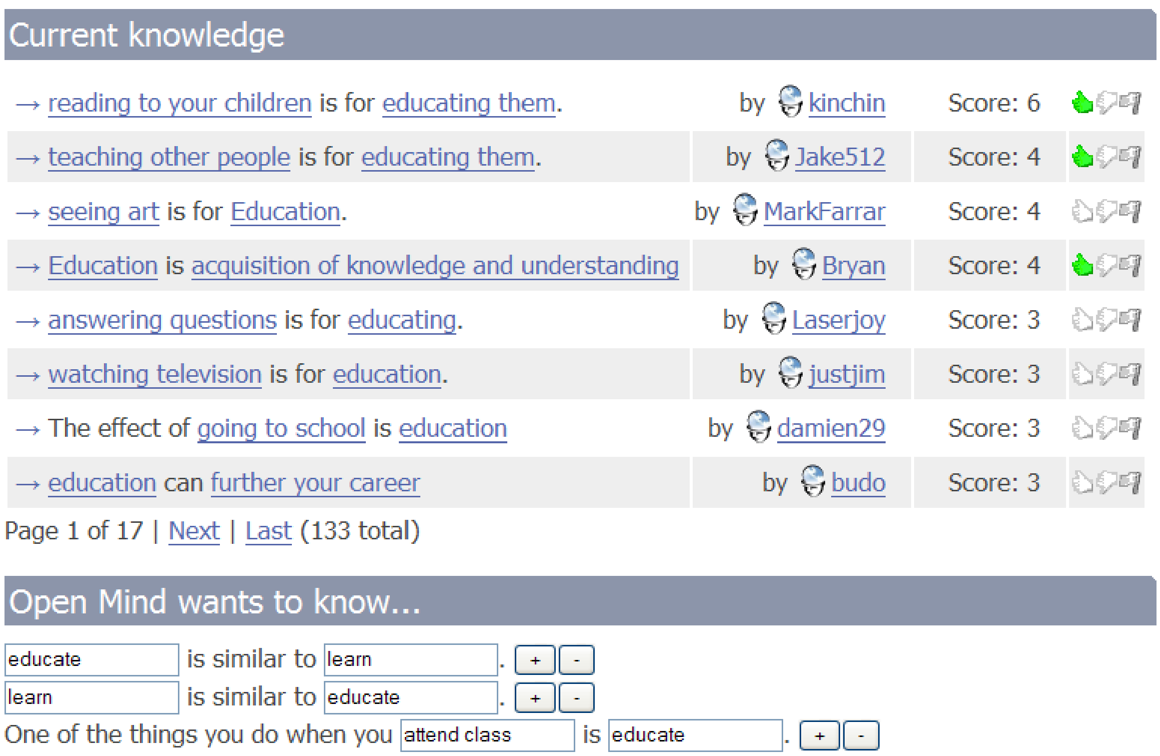


Figure 3-2: Partial screenshot of OpenMind Commons collection interface

3.2.2 Notable Limitations of OMCS corpus

PerspectiveSpace, when applied to the OMCS corpus, faces two kinds of sparse data problems. Efforts in data collection, up to this point, have not placed much emphasis

[4] For the purposes of PerspectiveSpace, a flag is treated as a neutral rating.

on encouraging contributors to rate existing assertions, and the result is that many assertions in the corpus have only the rating of the user who contributed the assertion. Furthermore, there is little controversy in the knowledge that has been collected: people tend to agree with most of the knowledge as "common sense," making it very difficult to find divisive issues that can distinguish groups of people and their belief structures.

At the same time, it is very important to note that these limitations are not inherent in the data structures underlying OMCS or even in the general mission of OMCS. Rather, these limitations are a consequence of the manner in which the data was collected, and efforts are underway to improve data collection. In the meanwhile, PerspectiveSpace is being tested on and applied to different and new corpora.

It should also be noted that, even with these limitations, PerspectiveSpace was able to yield impressive results in the more limited application domain of spam and abusive user amelioration.

Chapter 4

Approaches to applications

PerspectiveSpace opens user modeling and belief patterns to meaningful manipulations using elementary vector math. The applications built on PerspectiveSpace are likely to rely on a handful of basic methods.

4.1 Subculture detection

Each end of an axis of PerspectiveSpace (that is, of a principle perspective, as defined in 3.1.4) represents a group of people and the set of assertions (beliefs) those people agree to be true. On the opposite end of the axis are an opposing group of people and set of assertions (the inverse of the principle perspective). This is illustrated casually in figure 1-1. The use of SVD to calculate PerspectiveSpace yields a ranking of axes that dominate the data set. In other words, the axes of PerspectiveSpace represent the major divisions of belief patterns in the underlying data set. For example, people can be placed on an axis where one end has people who are fiscally conservative while the other end has people who are fiscally liberal. A distinct axis, however, could show a person's preference for cats over dogs or vice versa.

An alternative to exploring axes for subculture detection is to explore the person similarity matrix. This matrix is constructed by calculating UU^T, where U is the SVD component yielded in equation 3.3. The similarity matrix contains person agreement coefficients, which are a measurement of proximity between each person.

This matrix can then be analyzed by calculating a min-graph-cut or any other reasonable clustering over the person agreement coefficients. Such an analysis should readily recognize distinct cultural groups in the underlying data set, though it loses the orthogonality of perspectives: whereas a survey of axes can describe a person in terms of a sum of distinct belief patterns to which he or she may be subscribed, a min-graph-cut will only find the most major balkanizations of the people in the data set, without guaranteeing a description of the cause of the divisions.

4.2 Predictive ratings

Predictive ratings are easy to calculate, as the method is the same as AnalogySpace predictions. To predict the rating a person would give an assertion under the basic model of matrix construction (see 3.1.2), simply calculate $\mathbf{f}_P(i) \bullet \mathbf{f}_A(j)$. The resulting score represents, loosely speaking, the tendency of person i to agree with the people who tend to agree with assertion j.

Under the tagged model of matrix construction, the expression need only be updated to accommodate the additional tag term. To predict the rating a person i would give tag k on statement j, simply calculate $\mathbf{f}_P(i, k) \bullet \mathbf{f}_A(j)$.

It should be noted that these predictions are elements of $\hat{M}_R$, which approximates M_R. As such, even if there is evidence that a particular rating should be expected even though no rating was actually given, the value in $\hat{M}_R$ will likely tend towards 0, and so the magnitude of the predicted rating will often be significantly smaller than the magnitude of a value in $\hat{M}_R$ for which there is a nonzero value in M_R. A similar bias may be found for predictive ratings in $\hat{M}_R$ for which a nonzero value is given in M_R. Such biasing would not be present or as significant in a PerspectiveSpace generated using an SVD variant that treats missing values in M_R as unknown rather than zero.

Given the biasing effects inherent in predictive ratings and the SVD implementation used by this author, it is recommended that predictive ratings be used to compare values in $\hat{M}_R$ that have the same value in M_R.

4.3 Ad-hoc perspectives

Much like the ad-hoc categories of AnalogySpace [30], wherein a few examples of concepts belonging to a category are used to extrapolate other members of the category, ad-hoc perspectives in PerspectiveSpace use examples of people and/or assertions representative of a perspective to identify other people or beliefs that belong to the perspective.

At the heart of this method is the construction of the weighted sum (usually evenly-weighted) of person and/or assertion vectors corresponding to persons in the perspective. This resulting vector may be treated as either a person vector or an assertion vector in a predictive rating. For a person set S_p with a series of weight coefficients $\{a_i\}$ and an assertion set S_a with a series of weight coefficients $\{b_j\}$, a perspective vector can be computed as in equation 4.1:

$$\mathbf{v} = \sum_{i \in S_p} a_i \mathbf{f}_P(i) + \sum_{j \in S_a} b_i \mathbf{f}_A(j) \tag{4.1}$$

It should be noted that, to keep the magnitude of any predicted rating on the same scale as the values of M_R when using an ad-hoc perspective, the values of $\{a_i\}$ and $\{b_j\}$ should be normalized so that the following condition is met:

$$\sum_{i \in S_p} a_i^2 + \sum_{j \in S_a} b_i^2 = 1 \tag{4.2}$$

It should be noted that, although the i index in the equations of this section correspond to the person index in the basic model of matrix construction, this index can be taken to be an (i, k) pair, where $k \in \mathbb{P}_T$ under the tagged model.

4.4 Perspective projection

Perhaps one of the most powerful capabilities of PerspectiveSpace, perspective projection is the calculation of the set of assertions that would be perceived to be true by the people belonging to a particular perspective.

4.4.1 Decision, acceptance, and rejection

Any particular perspective, however, does not necessarily make a decision for every assertion in $\mathbb{P}_A$. The set of assertions D_a *decided* by a certain perspective vector $\mathbf{v}$ with threshold $\tau > 0$ is given by:

$$D_a(\mathbf{v}, \tau) = \{j | j \in \mathbb{P}_A, |\mathbf{v} \bullet \mathbf{f}_A(j)| > \tau\} \tag{4.3}$$

The set of assertions S_a *accepted* by a certain perspective vector $\mathbf{v}$ with threshold $\tau > 0$ is given by:

$$S_a(\mathbf{v}, \tau) = \{j | j \in \mathbb{P}_A, \mathbf{v} \bullet \mathbf{f}_A(j) > \tau\} \tag{4.4}$$

Similarly, the set of assertions $\bar{S}_a$ *rejected* by a certain perspective vector $\mathbf{v}$ with threshold $\tau > 0$ is given by either of these equations, equivalently:

$$\bar{S}_a(\mathbf{v}, \tau) = D_a(\mathbf{v}, \tau) - S_a(\mathbf{v}, \tau) \tag{4.5}$$

$$\bar{S}_a(\mathbf{v}, \tau) = \{j | j \in \mathbb{P}_A, \mathbf{v} \bullet \mathbf{f}_A(j) < -\tau\} \tag{4.6}$$

4.4.2 Tau selection

Care must be taken to select a proper threshold $\tau > 0$ to decide whether a belief belongs to a perspective or not. If an application requires the decision-making process of perspective projection over ranking solutions that can be obtained from using predictive ratings alone, it is generally advisable that a τ be chosen experimentally after some reasonable amount of data has been collected, and the choice of this threshold should be re-evaluated periodically. There are, however, some automated methods for choosing τ that may be considered.

Constant τ

It is the experience of this author that 10% of the magnitude of the score assigned to an explicit rating is reasonable, but this is very much affected by a number of highly variable factors, like the density of M_R. An experimenter building many different PerspectiveSpaces may find a suitable constant τ for his or her general family of applications. Otherwise, this method of choice is recommended for bootstrapping purposes only.

Leave-one-out estimation

The principle behind leave-one-out estimation is to determine, given a data set, what the approximate level a rating has in $\hat{M}_R$ when it is given a value of zero in M_R artificially and it is known that the rating should be given a non-zero score. This method can be computationally expensive.

For a single rating (i, j) with score s (that is, $s = M_R[i, j]$ and $|M_R[i, j]| > 0$), one could construct a matrix M'_R equivalent to M_R except that $M'_R[i, j] = 0$. Then, computing PerspectiveSpace for M'_R, an estimate for τ is given by $|\hat{M}'_R[i, j]|$ *provided* that the sign of $\hat{M}'_R[i, j]$ is the same as $M_R[i, j]$. If the sign is not the same, the estimate is invalid.

For some subset L of $\{(i, j)||M_R[i, j]| > 0\}$, create a distribution of valid τ-estimates T. A reasonable τ would likely be the first quartile or a low percentile (like the 10th) of that distribution.

4.5 Time sensitivity

People's tastes, preferences, and opinions change over time. Though this fact may be obvious in consideration of individuals, there are interesting implications for the analysis of the evolution of subcultures, which occur on a larger scale.

By replacing each user in the computation of a PerspectiveSpace with a representation of that user at a particular point in time, it would be possible to discover the propagation of beliefs through different groups.

In particular, this would mean that it is possible to detect how fads, trends, and other manifestations of opinion spread through populations, revealing implicit social networks. It would also be possible to chart the movement of a person through PerspectiveSpace over time, which has applications in education by helping understand students. This will be discussed further in section 5.5.

Chapter 5

Applications

The potential application domains for PerspectiveSpace, at least those contemplated by this author, fall into four major categories: characterizing societies, detecting microtheories, community-driven content, and recommender systems.

5.1 Characterizing societies

PerspectiveSpace offers many opportunities for the identification and study of small groups of people within a larger society. In particular, it offers a means for the systematic study of jargon usage, dialects, and belief patterns related to culture or subculture. In more general terms, PerspectiveSpace is a tool for opinion analysis, which has direct applications in marketing and political settings. With care, these "opinions" can be applied toward understanding the language of particular groups of people (or characterizing people by the language they use) in natural language processing applications.

5.1.1 Jargon and dialect analysis

As discussed in section 2.1.1, the OMCS corpus, by means of ConceptNet, has been increasingly successful at showing results of linguistic significance, with the general implication that language is a form of common sense knowledge. Intuitively speaking, it should be possible to use PerspectiveSpace and its capacity to isolate belief patterns to isolate basic patterns in language usage, including idioms.

To do so, proper framing[1] should allow people to distinguish their backgrounds by the terminology used in their statements. Consider again the following statements from section 3.1.1:

1. *Soda* is a carbonated beverage.

2. *Pop* is a carbonated beverage.

As such, PerspectiveSpace techniques, like subculture detection as introduced in section 4.1, should reveal localized uses of language like local idioms, field-specific jargon, and variations in dialect.

Natural language processing applications can use PerspectiveSpace to help model the language patterns of people dynamically by generating perspectives representative of a speaker from the language he or she uses, including jargon and idioms. If the PerspectiveSpace were constructed from a semantic resource, like OMCS, these perspectives could be used in later processing steps to aid with understanding the semantics underlying a statement. Given the example sentences above, a system can learn that a particular group of people using the word "pop" might be significantly more likely than other groups to mean a carbonated beverage rather than a small explosive sound.

5.1.2 SlantExplorer

SlantExplorer, which is covered in detail in chapter 7, is a web-based interface for navigating the conflicting opinions that underlie or are otherwise applicable to a document. It is designed as a tool for composing expository documents or for assisting users trying to summarize long documents.

SlantExplorer, for example, permits a user to explore a document on the ethics of abortion and see how the opinions of different groups of people are represented or

[1] See section 3.1.1 for a discussion of this subtlety

contradicted in different parts of the document. Similarly, a user can use SlantExplorer to model the reaction of a particular group of people to each segment of the document and see which segments are likely to be of concern to that group.

5.1.3 Understanding loan investment strategies

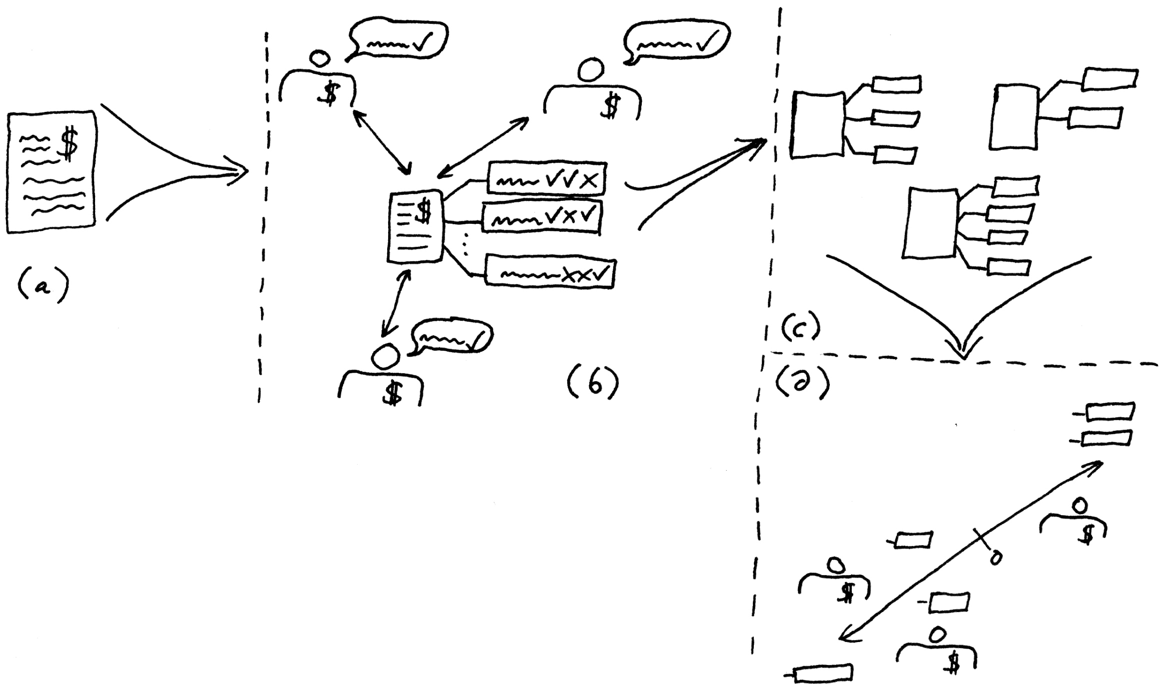
A bank issuing loans generally has several loan agents that each maintain a portfolio of loan instruments that he or she has purchased on behalf of the bank. A loan application then appears before one or more of these agents for consideration. A bank wishing to understand, quantitatively, the different loan investment strategies of its agents can use PerspectiveSpace to compare agents by their styles rather than by performance.

One configuration that can be used is shown in figure 5-1. Here, loan applications are reviewed by a handful of agents who each make a few short statements about each loan. Then, other agents can contribute more comments and/or agree or disagree with some of the existing comments. After a reasonable amount of data has been collected, a PerspectiveSpace can be computed over the data set, and this can be analyzed for subcultures as described in section 4.1.

The discovered subcultures should indicate the underlying belief patterns of the loan agents themselves. Coupled with an understanding of the assertions the agents are making and possibly disputing, this should reveal salient information on the investment strategies of these agents.

5.2 Preparation of microtheories

One particularly interesting feature of principle perspectives is that they likely contain assertions that are related to and are consistent with each other. This follows from the definition of the process of their discovery, as they are generally accepted or rejected in tandem by groups of people. As such, these assertions would constitute "microtheories."



(a) A loan application is selected to be a seed for analysis.

(b) The loan agents to be studied share their thoughts on the loan application, expressing their agreement or disagreement with each other (depending on the workplace community, this may require blind review).

(c) The loan application, with all of its attached assertions and reviews of those assertions, joins a larger pool for preparing a PerspectiveSpace.

(d) Subculture detection produces a report wherein loan agents and their assertions about loan applications are organized along dominant axes.

Figure 5-1: Model of an analysis of loan investment strategies

Microtheories are logically-consistent subsets of larger bodies of assertions, usually descriptive of a specific domain of knowledge. They are usually understood and discussed in the context of formal reasoning systems: the Cyc project, for example, is a body of common sense knowledge professionally-crafted into the form of formal, logical assertions [19]. In the field of informal reasoning in Artificial Intelligence, sometimes known casually as "scruffy AI," the automated recognition of microtheories is conspicuously missing. PerspectiveSpace, however, has the ability to expose microtheories without requiring machine understanding of the underlying assertions.

AI research into informal reasoning methods, particularly that which involves processing natural language (e.g. Wikipedia) into machine-usable knowledge, can use PerspectiveSpace-found microtheories to maintain coherence. That is, natural language resources are ridden with conflicting ideas and viewpoints, which can be prohibitively difficult to reconcile manually on a large scale. PerspectiveSpace, however, offers a means to reduce the reconciliation task, possibly to the point that other automated methods can complete it. As such, PerspectiveSpace lays a foundation for developing AI systems that can coherently maintain a viewpoint akin to the way people maintain their own opinions.

The limitation of using PerspectiveSpace to find microtheories is that the discovered microtheories are, by necessity of the discovery process, only microtheories that people accept or reject in groups. This contrasts with organized microtheories like Cyc, which generally use microtheories as groupings to limit access to only relevant assertions when trying to reason in a particular domain. In the context of the OMCS body of knowledge, this limitation would mean that each microtheory would be a controversial set of assertions, as would be seen in a debate over the ethics of abortion or the acceptance of Darwin's theory of evolution. There are practical applications surrounding this limitation, and a prototype of such an application, SlantExplorer, is discussed in chapter 7.

In order to understand a principal perspective's microtheory, it is crucial that the beliefs of the perspective be examined to identify the trend or property measured by the principal perspective. This is how the axes in AnalogySpace, including the axes of "desirability" and "feasibility," were discovered in the OMCS data set [30].

5.3 Community-driven content

PerspectiveSpace offers a variety of interesting possibilities for applications that use community-driven content. Tested and proposed uses in the Common Sense Computing Initiative's own OpenMind Common Sense project are considered in this section, followed by a review of some possibilities for two well-known community driven projects: Slashdot [26], and Wikipedia [34].

These three projects share the feature that people can contribute to and navigate a body of knowledge on which not every user would agree. The ability to let users regulate their exposure to incompatible perspectives should enhance the experience of contributors and users alike in these projects.

5.3.1 Perspective projection and subculture detection for contributors and browsers

The use of perspective projection, as described in section 4.4, can allow contributors and browsers of a community-driven content project to work in a space of assertions that are compatible with their particular belief patterns. For a contributor, this would allow him or her to build upon existing statements in greater detail and comfort. For a browser, this would allow him or her to explore content in a self-consistent form.

Subculture detection, on the other hand, can help a contributor or browser identify and understand the major sides of an argument as well as the prevailing agreements underlying a discussion.

Used in tandem, perspective projection and subculture detection can help either user class manageably explore conflicting perspectives for considering counter arguments or differing opinions.

Perspective projection in OpenMind Commons

It is in the spirit of the mission of the OMCS project to allow volunteers to define common sense knowledge for themselves, leaving it to the project maintainers to find ways to distill machine-usable information from the contributions. Unfortunately, not all contributors can agree with each other, and some contributors are abusive, aiming to deface the project. It is not in the interest of OMCS, however, for the project maintainers to screen for "legitimate" contributors explicitly, as this would be costly and impose experimenter bias on the collected data. Perspective projection is an efficient and effective alternative.

Implications of PerspectiveSpace in Slashdot

Slashdot [26] is a very active online news and discussion forum serving a community of highly computer-literate experts and users. Given Slashdot's level of activity, users are presented with the challenge of finding worthwhile content in the face of information overload. To help address this problem, Slashdot and its user community developed a moderation system whereby users rate each other's comments (sometimes applying tags like "insightful"), a praxis for its usage, and a content filtering mechanism built on these ratings [17].

Slashdot's moderation system is very elaborate in that it has a mechanism for awarding moderation points, an additional mechanism for moderating moderators, etc. Users see the total accumulated ratings for each comment. Users rarely adjust their personal preferences with respect to the filtering system [18]. Given this and the way the collected rating data are handled, Slashdot can be understood as a forum designed to cater to a user base dominated by a single harmonious community.

PerspectiveSpace could contribute to this community, or any community of similar structure, by allowing users to organize themselves and find content suited to their particular tastes with the same degree of effort, or perhaps even less, that is required of them under Slashdot's present system. Perspective projection could very easily let users find only the comments that they consider insightful. Subculture detection could provide a service similar to SlantExplorer by allowing users to identify, at a glance, the major sides of an argument or the prevailing perspective of a discussion.

Implications of PerspectiveSpace in Wikipedia

Wikipedia [34] is a very successful project built on computer-mediated communication (CMC), wherein visitors to a website in a wiki format can browse and directly edit a community-built encyclopedia. Given the enormous and growing size of the Wikipedia user community, the people who contribute to the project have varied backgrounds, opinions, and goals (some of these goals are not in line with the mission of the project itself, as is manifested in vandalism). As noted in a comprehensive study across all Wikipedia articles and their histories [16], a growing portion of Wikipedia edits take the form of conflict and coordination. Despite this increase in effort, the portion of articles that could be reasonably considered "damaged" is small but increasing [23].

A team of researchers from UCLA and PARC [16] developed models for understanding Wikipedia conflict at three distinct levels: global, article, and user. For their user-level model, they built a system called Revert Graph, which uses a simulated annealing algorithm to cluster the users who have edited a given article into disparate camps that represent revert conflicts among the camps. It should be note that, under this model, a user cannot belong to more than one camp, and each analysis is applicable only to a single article.

Another project introduces the concept of Controversy Rank [32], which establishes an automated metric for identifying controversial articles in Wikipedia. Projects such as this and the work at UCLA and PARC do excellent work in identifying controversies and even make good observations into characterizing them, but little work is done for navigating them or resolving them to the satisfaction of individual users.

PerspectiveSpace can contribute to this field of work by permitting an analysis that spans all Wikipedia articles simultaneously while teasing out the different aspects of a user's perspective, rather than assigning each user to only a single camp. In doing so, browsers can be shown, by means of perspective projection, the revision of any given article that is most compatible with their own perspective. This does not preclude the presentation of conflicting ideas, however, as users can be offered, in an organized fashion, the competing perspectives that were otherwise hidden. For articles where disputes are ongoing, this could have the potential benefit of maintaining coherence within an article for a browser.

Artificial Intelligence research seeking to use Wikipedia as a knowledge corpus may find subculture detection unto itself useful as well. If articles can be broken into disputed pieces, subculture detection should reveal self-consistent microtheories[2]. AI research can then build reasoning models conscientious of controversy and the assertions that comprise any given perspective.

5.3.2 Spam detection and/or amelioration

Spammers who work by methodically entering bad data can be detected by examining the user similarity matrix, where they should appear with a negative coefficient in comparison to the bulk of the contributors. Spammers who work by using multiple accounts to rate spam assertions in unison can be readily detected using the subculture detection mechanism already described.

It is important to note that perspective projection automatically eases the effect of spam contributors. In such a system, spam users would operate in a self-consistent/spam-consistent collection of community-driven content, while the rest of the user body would never see those assertions for the most part.

Possible attacks

There are two elementary ways in which PerspectiveSpace can be thwarted so as to give the beliefs of malicious users unfair visibility, which is a concern especially in those situations where PerspectiveSpace was chosen to help alleviate such problems.

Partial agreement attack Since many community-driven content projects, including Wikipedia and OMCS, open their data sets for public download, spammers

[2]See section 7.2.2 for an example of subculture detection revealing microtheories.

or otherwise abusive users can download these data sets to manipulate their positions in PerspectiveSpace. As such, a user attempting to defeat the organization of PerspectiveSpace to inject beliefs into a particular perspective can impersonate the agreement decisions that would be made by people rightfully belonging to that perspective. Once the attacker has artificially established his/her position in PerspectiveSpace, the attacker can make or rate assertions according to an agenda distinct from the PerspectiveSpace application. This type of attack would be a "partial agreement attack."

Flood attack Furthermore, a massive injection of malicious data could, theoretically, push legitimate data into the minority. Simplistic PerspectiveSpace-based spam detection and amelioration implementations based on majority rule would then invert their effects, hiding legitimate contributions and promoting malicious ones. This would be a "flood attack."

As such, several measures are required in conjunction with PerspectiveSpace in a spam detection or amelioration application.

Simple countermeasures

CAPTCHA A simple measure would be to limit the number of malicious accounts that can be created and controlled in synchrony. This is a classic problem that can be addressed with a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [31], like reCAPTCHA [6]. This should effectively prevent any one attacker from overwhelming the data set with a flood attack.

Rating classes Another measure, as mentioned in section 3.1.2 is to modulate the magnitude of the values going into the matrix M_R by dividing ratings into explicit and implicit rating classes. If implicit ratings are rated more highly than explicit ratings, then that means an abusive user attempting the partial agreement attack would find his/her position in PerspectiveSpace dominated more by the user's contributed assertions (in this case, the injected assertions) than by the explicit ratings originally used to establish the user's position.

Time-limited scoring A more aggressive countermeasure to accomplish the same effect would be to reduce the score assigned to each explicit rating given by a user on any given day as the number of ratings given by that user on that day increases. Automated attempts would thus be even more weakly-positioned in terms of the perspective that was supposed to be impersonated, and be even more strongly-positioned by the assertions that were injected into the system. If the PerspectiveSpace application is designed to maintain a long-term relationship with its contributors, as might be reasonably expected of such a system, then regular contributors are not likely to be adversely affected by the modulation.

Aggressive normalization If the unity-magnitude algorithm, as described in section 3.1.3, is used as part of the matrix normalization process, then the aggressive choice of a base parameter can dampen the effects of explicit ratings even further. This should be considered only in conjunction to the time-limited scoring scheme discussed above, as a very aggressive base parameter would easily dampen the effects of explicit ratings from legitimate contributors.

Defense against partial agreement attacks must generally be a careful balance of collection interface design (given the behavioral patterns of legitimate contributors) and fine-tuning the matrix preparation and normalization rules leading up to the calculation of $\hat{M}_R$.

A discussion of the first implementation of PerspectiveSpace, which was created to combat abusive user behavior in OMCS, is given in chapter 6.

5.4 Recommender systems

Perhaps one of the most direct applications of PerspectiveSpace is in recommender systems. Predictive ratings, which would otherwise be substantially similar to current best-practice methods for recommender systems, have the unique ability to base similarity of opinions on the *way people think* about the items in question rather than just the *way people shop* or the undifferentiated “star rating” assigned to an item.

An important distinction between traditional recommender systems and the role that PerspectiveSpace can play is that traditional recommender systems suggest a particular product or item that would likely be desirable, whereas PerspectiveSpace can suggest agreeable opinions on products and items, leaving the determination of desirability open to human interpretation. Granted, the methods of computation are substantially similar, but a proper PerspectiveSpace implementation can easily collect more data than a traditional recommender system. With this extra data, PerspectiveSpace can produce reasonable results for recommending opinions of entities that would traditionally be recommended themselves.

For applications that demand automated filtration for desirability, a system can make use of specially-marked opinions of desirability. 2-wit’s data model provides for “stock assertions” for this purpose, though they have not yet been put into use at the time of this writing. These stock assertions can capture information stored in traditional rating systems, like the number of “stars” a review would assign to an entity. More carefully-chosen semantics for stock assertions could permit reasoning over the types of beliefs held about an entity. In this circumstance, PerspectiveSpace would play the role of a traditional recommender system, but PerspectiveSpace would draw upon the much richer data set of specific opinions on products to make its recommendations.

2-wit (an abbreviation of “What Will I Think”) is a PerspectiveSpace implementation and application designed for consumers. 2-wit permits users to enter free-text opinions about various entities, review opinions entered by other users, and tag opinions as “agreeable,” “disagreeable,” or “junk.” When reviewing opinions entered by other users, users can see assertions prioritized according to how likely they are to agree with them and filtered according to whether or not they would likely consider them spam, obscene, or otherwise inappropriate.

So far, 2-wit accepts only movies as valid entities, but it is designed to expand naturally to include restaurants. It is also the expectation of this author that 2-wit can be extended to include consumer products.

5.5 Education

It has been shown that a common sense knowledge repository can assist with education, particularly by identifying common misconceptions and common foundations for learning new ideas [7, 3]. PerspectiveSpace can improve upon these approaches by being sensitive to the particularities of the individual learners.

Since PerspectiveSpace can assist directly with common sense applications in education as well as lending its own unique methods, what follows here is a survey of methods in which common sense knowledge can be used in the preparation of learning activities before a detailed discussion of PerspectiveSpace's applicability.

5.5.1 Applicability of common sense

Common sense knowledge has been shown in [7] to help teachers in preparing learning activities. The following is an excerpt taken from that paper, showing some of the preparation steps where common sense can help:

1. identify topics of general interest to be taught;
2. identify student misconceptions that are inappropriate in a certain context;
3. fit the instructional material to the learner's previous knowledge; and
4. provide a suitable language to be used in the instructional material. [7]

The general principle underlying the use of a common sense knowledge base in teaching is that it can represent the "student model" (or "naïve model"). This is a model of what the learner is expected to know already or otherwise believe, whether correct or not. The work in [3] collected demographic information from contributors, so that the collected knowledge can be filtered to obtain a representative sample for a target demographic.

PerspectiveSpace can improve on this model by allowing a teacher to generate, by means of perspective projection, student models representative of the actual students at hand, rather than relying exclusively on an approximation determined by the relevant demographic.

Identification of materials to cover

By studying the contents of the student model of knowledge surrounding a particular topic of interest, a teacher can identify topics of interest for the particular demographic (by looking at the relative proportion of assertions surrounding a topic), misconceptions (by identifying common myths and fallacies in the data set), and ignorance (by noting assertions that are altogether missing).

A perspective projection of a student's knowledge would require data from the student revealing much of what he or she does and does not understand, but PerspectiveSpace would also contribute an estimation of what concepts the student would likely understand or misunderstand in tandem. Indeed, by identifying the concepts that are usually understood together, a teacher can understand what supporting concepts are missing before new material can be properly understood by the student.

Structuring new materials

As examined in [7], effective learning requires that new knowledge be connected to existing knowledge in a learner's mind. The trivial example used in that paper is that a student learning that "banana is a nutritional fruit" may connect ideas from his or her knowledge of "bananas" to his or her knowledge of "fruits" and "nutrition." A student model can therefore assist a teacher by revealing the relevant concepts that are already understood by the learner.

In a similar vein, a student model can expose the vocabulary and other features of language that are best understood by the learners. It has been shown in [13] and ongoing research that language features can be gleaned from the ConceptNet representation of common sense knowledge. When it comes to communicating ideas to learners, however, Kumar and Lieberman demonstrated a system called SuggestDesk that uses the ConceptNet resource to discover common sense analogies that a domain expert can use to help communicate ideas to the average person [20].

PerspectiveSpace can improve the student model used by these approaches by preparing an approximation of the model for a particular student or group of students using less than a complete diagnostic evaluation of each student. With sufficient data, this improved model would likely include even the nuances of beliefs in the students' social groups.

5.5.2 Unique contributions of PerspectiveSpace

The possible contributions of PerspectiveSpace to education become particularly interesting when considered with time sensitivity (see section 4.5. It is possible to study the movement of students through PerspectiveSpace as they learn, therefore revealing likely hurdles for future students even when they have not been exposed to the material in question. That is, it is possible to estimate future misconceptions as indicated by past students with a similar mindset. At the same time, it is possible to see the course of changes in similar students in the past as they overcame a learning hurdle, which can be used by a teacher to improve learning activities in the future.

5.5.3 Requirements

PerspectiveSpace requires the use of extensive data collection, but the use of an electronic testing paradigm (especially diagnostic tests) can provide the data PerspectiveSpace needs to determine what pieces of knowledge tend to fit together.

Chapter 6

Combating abusive users in OMCS

6.1 Background

Not long ago, a story about one of the Common Sense Computing Initiative's early founders appeared in a popular technology magazine, including a link to Commons, the current common sense elicitation site for OMCS. This provided a normally welcome publicity boost to the project, but it attracted the attention of an abusive user. This user selected a legitimate contributor to target, whom he proceeded to attack systematically by:

1. rating all of the contributor's assertions down (see figure 6-1),

2. making multiple obscene assertions about the contributor (see figure 6-2), and

3. creating over 60 accounts for the purpose of rating these assertions in unison.

The primary mechanism that Commons uses to determine the degree of truth for an assertion is the assertion's score, which is just the sum of all ratings on that assertion (+1 for an agreement, −1 for a disagreement, +0 for no rating). In this mechanism, all users are considered equal.

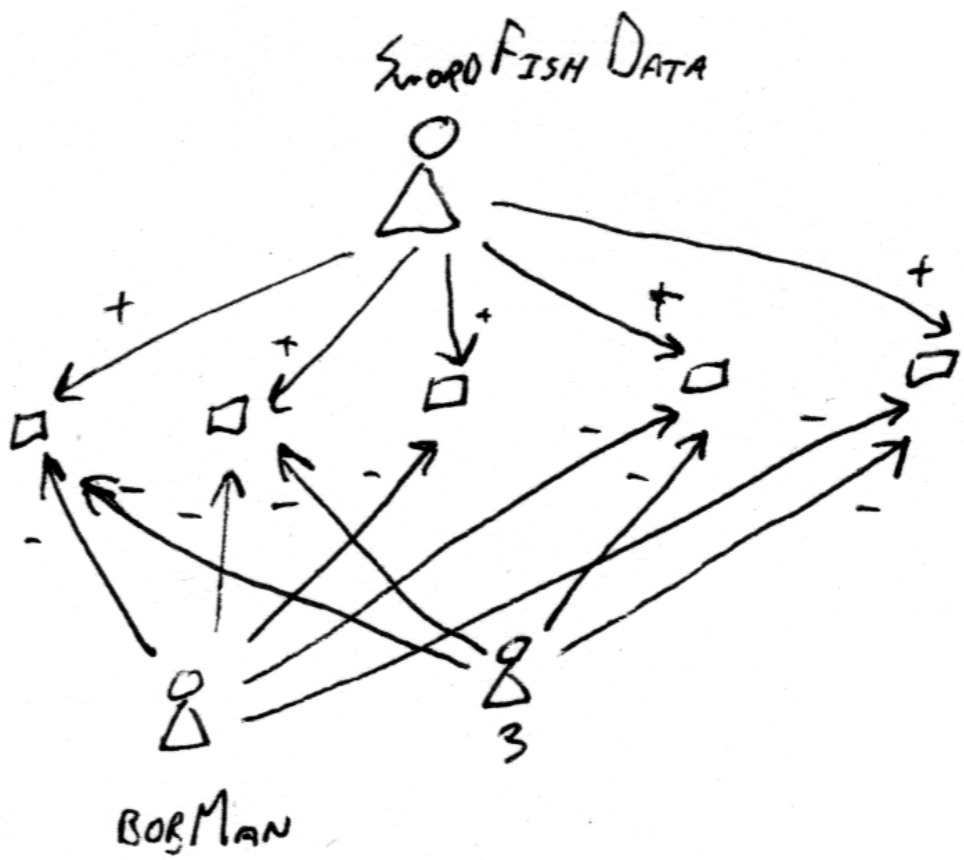


Figure 6-1: An illustration of the assertions made by a legitimate contributor (SwordFishData) being systematically rated down by an abusive user (bobMan) and one of his numbered drone accounts (3)

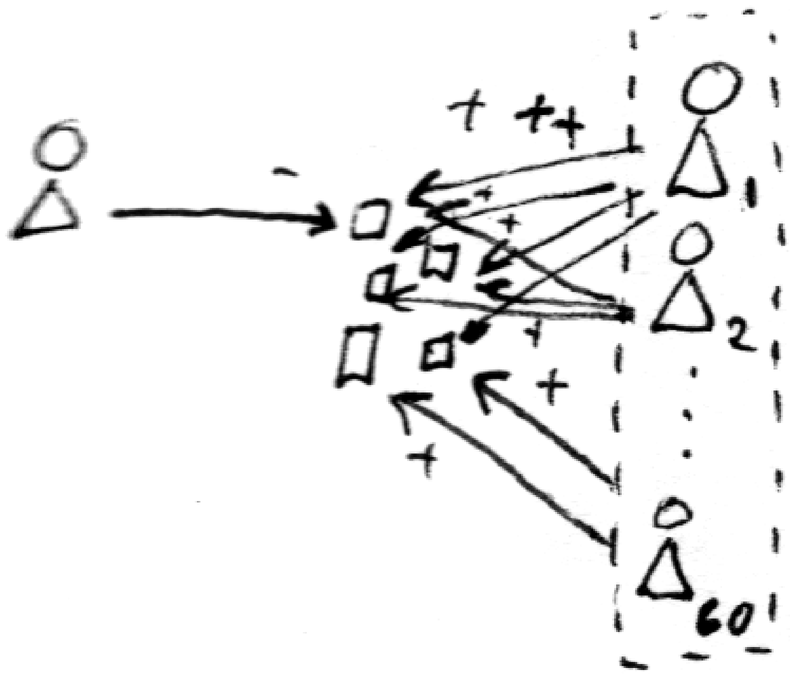


Figure 6-2: An illustration of a group of assertions (mostly profane in nature) being rated up by a large number of drone accounts, with one legitimate contributor rating down one assertion in the group

6.2 Construction of PerspectiveSpace

Given these patterns, it was clear that an SVD of ConceptNet's rating data (that is, the calculation of what would become the first PerspectiveSpace) should readily reveal the synchronized ratings and the dispute between two users.

The construction of PerspectiveSpace for this application went through several iterations. The first iteration involved no distinction between explicit and implicit ratings, no tags, and only unity normalization. The matrix construction rules are given in table 6.1, and the results are discussed in the next section.

Choice	Meaning	$M_R[i,j]$
None	No choice made	0
Thumbs-up	Agree with assertion	1.0
Thumbs-down	Disagree with assertion	-1.0
Flag	Fix grammar	0

Table 6.1: Semantics of the most basic construction of PerspectiveSpace over the OMCS data set

After mirroring, implicit ratings, and other advanced PerspectiveSpace concepts were theorized and implemented for 2-wit, another PerspectiveSpace over the OMCS data was constructed using these principles. The matrix construction rules for this analysis are given in table 6.2.

		$M_R[i,j]$	
Choice	Meaning	Explicit	Implicit
None	No choice made	0	0
Thumbs-up	Agree with assertion	1.0	2.0
Thumbs-down	Disagree with assertion	-1.0	-2.0[1]
Flag	Fix grammar	0	0

Table 6.2: Semantics of a construction of PerspectiveSpace over the OMCS data set, using explicit and implicit ratings

6.3 Observations on the incident

The PerspectiveSpace plot in figure 6-3 shows the bulk of all OMCS contributors, who made bona fide contributions, as a few comet-like streamers of points in PerspectiveSpace. The plot in figure 6-4 shows the same space, but it emphasizes how the abuser's many drone accounts[2] polarized the top three axes of the OMCS data set: the tight line of drone accounts in the negative octant indicates that the positions of drone accounts across these axes are strongly correlated, whereas the less-defined cloud of regular contributors in the positive octant indicates that the positions of regular contributors across these axes are less strongly correlated. More simply, this diagram can be interpreted as to show that the drone accounts uniformly contest the assertions of the bulk of all legitimate contributors, which can be divided into major groups that otherwise do not have any significant correlation with each other.

[1]The OMCS collection interface allows for a contributor to disagree with his/her own assertions. This behavior can be construed to be abusive, but it is rare and no effort has been made to give this circumstance special attention in the preparation of a PerspectiveSpace.

[2]The astute reader will notice that there appears to be a point labeled "SwordFishData" in the midst of the drone accounts, even though the abusive user tended to contradict that user, who tended to make bona fide contributions. The attacker created his own account called " SwordFishData" (with a leading space, which was generally not rendered in web browsers) for his own nefarious purposes, and that is the data point seen here.

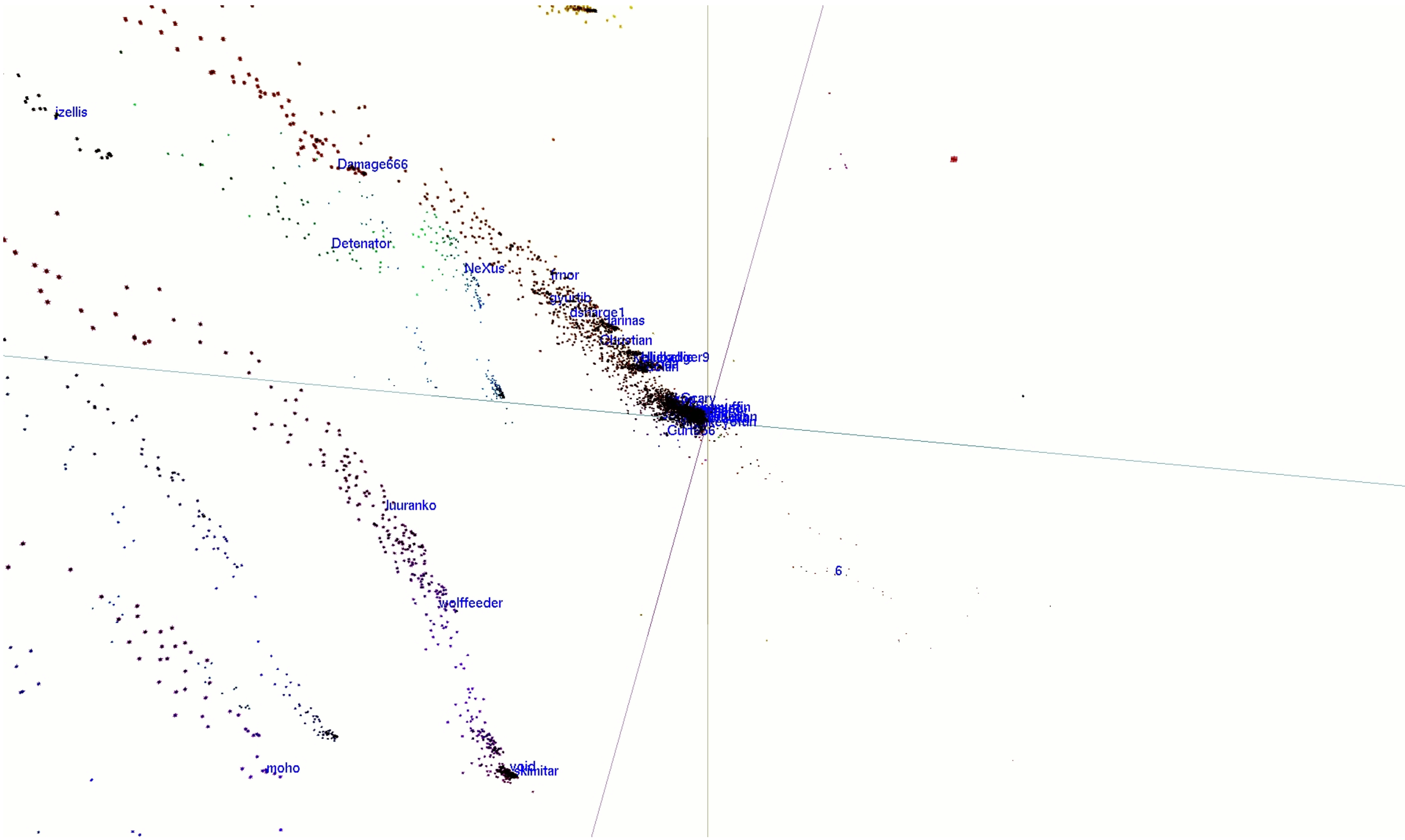


Figure 6-3: A plot of the top three axes of the OMCS PerspectiveSpace focusing on the bulk of bona fide contributors, appearing as a series of comet-like streamers

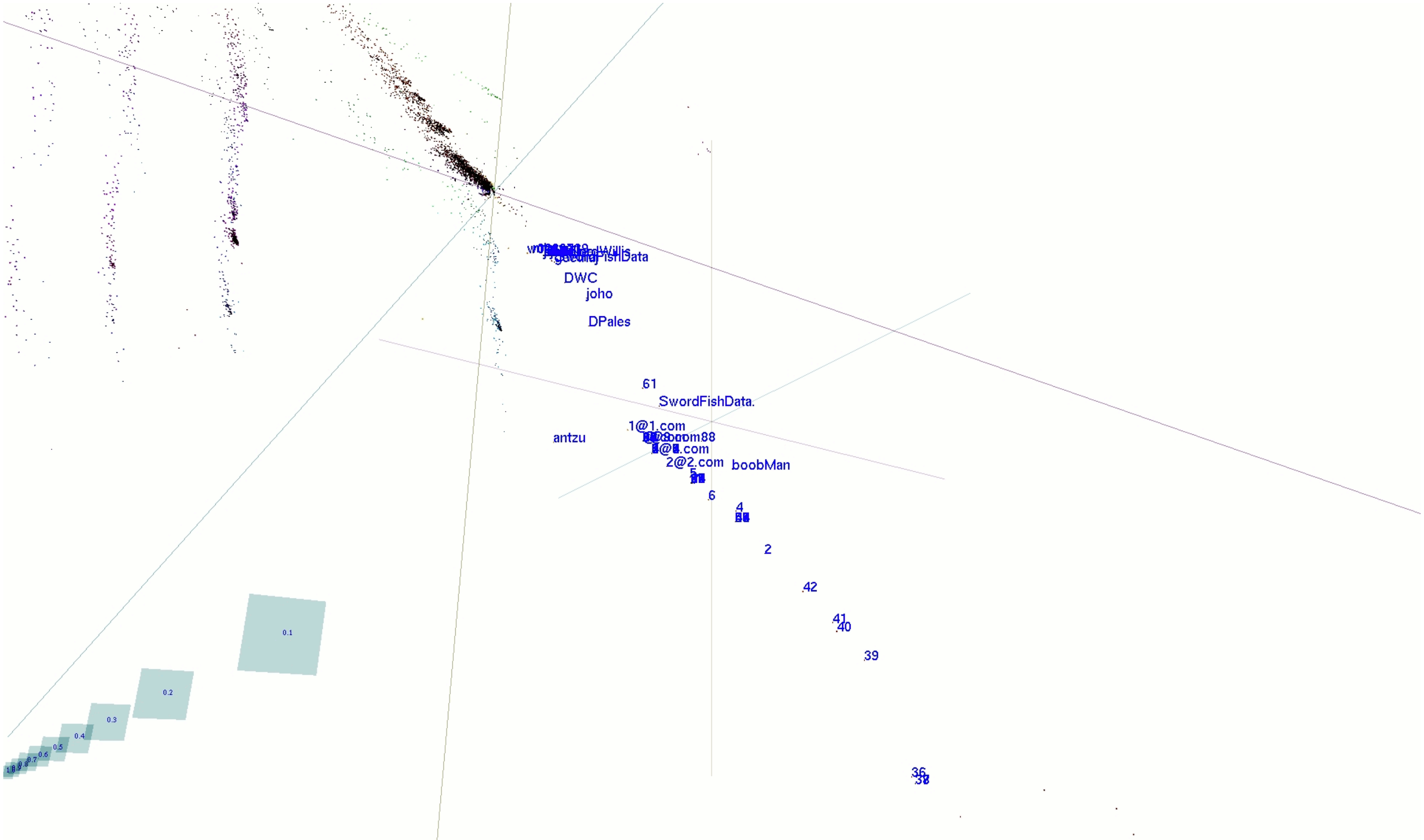


Figure 6-4: A plot of the top three axes of the OMCS PerspectiveSpace, now focusing on the long tail of abusive drone accounts and junk-contributors; note that these accounts stretch far away from the bona fide contributors along opposing axes

Chapter 7

SlantExplorer

SlantExplorer is a web-based interface for navigating the conflicting opinions that underlie or are otherwise applicable to a document. SlantExplorer is designed as a tool for composing expository documents or for assisting users trying to draft document summaries. By considering the perspectives that people can take on a document, SlantExplorer can help identify portions of a document of interest to people belonging to particular groups or backgrounds.

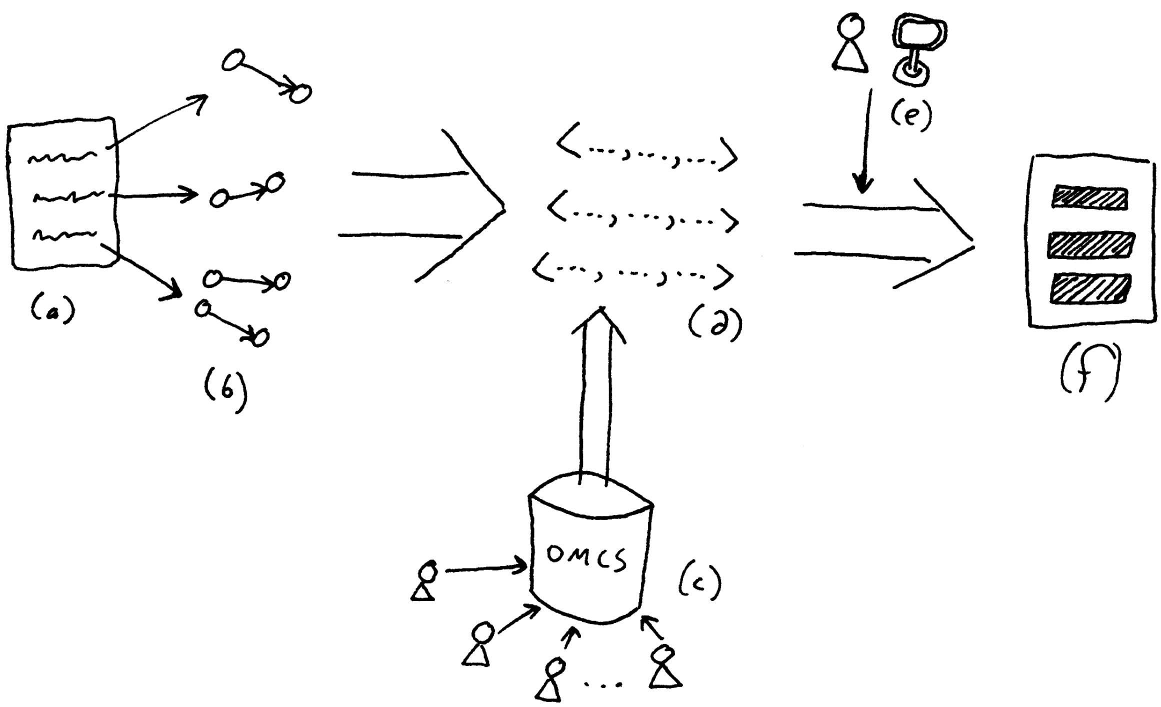
The ideal implementation of SlantExplorer requires advances in natural language processing technology otherwise outside of the scope of this research, though progress is being made on the relevant fronts. As such, this chapter covers the theoretical architecture of SlantExplorer first, and then there is a discussion of the implementation of a SlantExplorer prototype.

7.1 Architecture

The intended process for a running SlantExplorer implementation is illustrated in figure 7-1. An informal treatment of this design is given here, and a more rigorous treatment is given in the following subsections.

Analysis begins with loading the perspective vectors corresponding to the assertions underlying each segment of text. These vectors are assigned directly to each segment. If more than one vector would be assigned to the same segment through this method, all candidate vectors for a segment are summed to compute the final vector, weighting according to the relative significance of each assertion. If a segment has no vectors, it will not be subject to highlighting by SlantExplorer.

The algorithm for identifying the top axes for a document is elementary. For this calculation, each vector in the document is temporarily replaced with a vector whose components are the square of the components of the original vector. All of these vectors are them summed to produce a new vector whose components reflect the relative prominence of each PerspectiveSpace axis in the document. The axes corresponding to the largest few components of this vector are taken to be the default axes for describing the document.



(a) Source document is presented.

(b) A natural language processing tool finds one or more assertions underlying each sentence.

(c) The Open Mind Common Sense corpus, compiled into PerspectiveSpace, is consulted to find a perspective vector for each assertion.

(d) The vectors for the assertions for each sentence are combined to produce one perspective vector per sentence.

(e) The SlantExplorer user specifies the axes and/or ad-hoc perspectives of interest.

(f) The target axes and perspectives, combined with the sentence vectors, are used to annotate the original document for review by the user.

Figure 7-1: Architecture of an ideal SlantExplorer implementation

7.1.1 Segmentation

A source document on some matter of possible controversy is presented to the system by the user (the example used throughout this chapter is an article from Wikipedia [34] on the ethics of abortion). This document is divided into smaller segments of text, which will presumably be sentences, though other divisions may be acceptable or appropriate as circumstances demand and technology permits. These segments will be the smallest units of annotation throughout the course of SlantExplorer's processing, and the segment size should be chosen accordingly. For the purposes of this chapter, the sequence of segments shall be denoted as $\{s_1 \dots s_n\}$.

7.1.2 Understanding

A natural language understanding tool must then process the document to produce a set of assertions $A \subset \mathbb{P}_a$ underlying the statements in the document and find a coefficient of relevance $r_{i,j}$ between each segment s_i and assertion $a_j \in A$. The coefficients of relevance (denoted collectively as the matrix R and separately as the elements $r_{i,j}$) should have the following properties:

1. $r_{i,j} = 0$ iff segment s_i has no relationship with assertion a_j;

2. $r_{i,j} > 0$ iff segment s_i expresses agreement with assertion a_j;

3. $r_{i,j} < 0$ iff segment s_i expresses disagreement with assertion a_j; and

4. given any i for which there is at least one nonzero $r_{i,j}$, $\sum_j r_{i,j}^2 = 1$.

It should be noted that "expresses agreement" and "expresses disagreement" are vague terms. Different documents may cover controversial issues in different ways, and the SlantExplorer user may wish to perform an analysis particular to the method used. The Wikipedia article on the ethics of abortion, for example, is framed in mostly neutral language (that is, the document discusses different stances on an issue without taking a stance itself). One SlantExplorer user might want to analyze the document with respect to the stances discussed, while another might want to analyze the document with respect to the facts claimed, possibly seeking non-neutral language. To the extent that natural language processing technology permits, the SlantExplorer user should have discretion over the type of natural language understanding performed.

The final required property of $r_{i,j}$ is a special case of equation 4.2, which insures that the ad-hoc perspectives (see 4.3) created for each segment in the next step has the proper magnitude.

7.1.3 Perspective identification

Once the set of assertions A and coefficients R are found, the perspective vectors for each assertion a_i need to be loaded from a PerspectiveSpace preparation of a large corpus of human knowledge and opinions, like OMCS. These vectors are denoted as $\mathbf{f}_A(a_i)$.

Given these PerspectiveSpace vectors, the perspective represented by each segment, denoted as $\mathbf{p}_i$, is then calculated using the coefficients of relevance as a weighting term:

$$\mathbf{p}_i = \sum_j r_{i,j} \mathbf{f}_A(a_i) \tag{7.1}$$

It should be noted that this is a special case of equation 4.1 for defining an ad-hoc perspective, as defined in section 4.3.

7.1.4 Annotation

Once the perspective underlying each segment has been found, the user has three major possible requests for annotation:

1. reveal the dominant principal perspectives in the document;
2. reveal the dominant principal perspectives out of a few select perspectives in the document;
3. given a particular perspective (that is, from known groups of people and/or assertions, possibly derived from segments selected in a previous annotation), indicate sections of the document of interest.

Revealing dominant perspectives

For this activity, each segment is characterized in terms of the axis on which it is most prominently placed. If the k^{th} component of $\mathbf{p}_i$ is denoted as $p_{i,k}$, then the dominant axis d_i of $\mathbf{p}_i$ is given by:

$$d_i = \arg\max_k p_{i,k}^2 \tag{7.2}$$

Each unique value of d_i can correspond to a distinct coloration in a highlighting interface, which will be discussed in 7.2.3. However, interface limitations and legibility concerns are likely to restrict the number of represented axes to a select few, whereupon the most descriptive axes of the document must be selected.

Revealing dominant perspectives out of a select subset

In this activity, only a few particular axes are candidates for annotating segments. Oftentimes, this will be preferred over the method described in the previous section, as following a smaller number of axes will better expose the treatment of those particular axes throughout a document. This section discusses the automatic selection of descriptive axes of a document first.

Identifying the most descriptive axes of a document has important distinctions from identifying the perspective of a document. In particular, a document that expresses opinions on either side of a core issue (that is, an issue represented by an axis in PerspectiveSpace) may have a trivial position on that axis overall. Producing a vector that represents the degree of activity along each axis in a document requires a different procedure from perspective detection. A document's degree of activity can be described on a per-axis basis q_k for axis k as follows:

$$q_k = \sum_i p_{i,k}^2 \tag{7.3}$$

Once these degrees of activities are determined, the indices of the top m most descriptive axes as indicated by $\{q_1, q_2 \ldots\}$ can be represented by $\{k_1 \ldots k_m\}$. These indices, in turn, can be used to identify the most significant axis *that is relevant to the document as a whole* for each segment:

$$d_i = \underset{k \in \{k_1 \ldots k_m\}}{\arg\max}\, p_{i,k}^2 \tag{7.4}$$

Alternatively, the series $\{k_1 \ldots k_m\}$ can be supplied by the user.

Indicating segments of interest

For this activity, segments that are significantly in line with or opposed to one or more user-provided perspectives are indicated as such. Perspective projection, as described in 4.4, is the underlying technique for this analysis.

These perspectives can be provided in any of a number of ways: ad-hoc perspectives as described in section 4.3 (this includes composing a perspective out of known people and assertions), or from segments in a document that has been analyzed in a previous application of SlantExplorer. In either case, a user can prepare a perspective by figuratively dropping each person, assertion, or segment into a “for” or “against” bucket, optionally providing a weight to indicate the relative importance of each person, assertion, or segment in defining the perspective. However the perspective is obtained, that perspective shall be notated as $\mathbf{v}$ for this section, while the chosen threshold for accepting a decision on any given segment is τ.

With these parameters, it is possible to define functions for the decision, acceptance, and rejection of a segment by a given perspective in a form similar to equations 4.3, 4.4, and 4.6 from perspective projection.

The segments of interest $I_s(\mathbf{v}, \tau)$ to a perspective $\mathbf{v}$ is given by:

$$I_s(\mathbf{v}, \tau) = \{i \mid |\mathbf{v} \bullet \mathbf{p}_i| > \tau\} \tag{7.5}$$

The set of segments S_s *accepted* by a certain perspective vector $\mathbf{v}$ with threshold $\tau > 0$ is given by:

$$S_s(\mathbf{v}, \tau) = \{i \mid \mathbf{v} \bullet \mathbf{p}_i > \tau\} \tag{7.6}$$

Similarly, the set of segments $\bar{S}_s$ *rejected* by a certain perspective vector $\mathbf{v}$ with threshold $\tau > 0$ is given by either of these equations, equivalently:

$$\bar{S}_s(\mathbf{v}, \tau) = I_s(\mathbf{v}, \tau) - S_s(\mathbf{v}, \tau) \tag{7.7}$$

$$\bar{S}_s(\mathbf{v}, \tau) = \{i | \mathbf{v} \bullet \mathbf{p}_i < -\tau\} \tag{7.8}$$

7.2 Implementation

The prototype of the SlantExplorer concept could not realize the complete design as outlined in the previous section for two major reasons. The first is that readily-accessible natural language processing technology is not developed sufficiently at this time to process a body of text into assertions of the granularity ideal for inclusion in OMCS. It should be noted that there is existing work, a projected called Concept-Miner [8], that address the precise issue of extracting OMCS-compatible common sense knowledge from public sources, but it does not operate with the granularity required for SlantExplorer. The second reason is actually complementary: the OMCS common sense knowledge collection, as was discussed in section 3.2.2, is lacking in controversial opinions and a suitable mechanism for eliciting them as of the time of this writing.

This section is dedicated to explaining the compromises made to build a SlantExplorer prototype and the consequences thereof.

7.2.1 Substitute for natural language understanding

As a substitute for developing and/or integrating a natural language understanding system, this author instead opted to select a demonstration document for analysis and to prepare A and R by hand. Given the natural limitations of a human resource, this also means that a user of the SlantExplorer prototype does not have any discretion over the type of natural language understanding performed, which should otherwise be made available to the user as discussed in section 7.1.2.

The demonstration document chosen for the prototype was the English Wikipedia article on the ethics of abortion [33]. This document and its underlying issues were chosen for their understandability by the general public and the widespread and polarizing opinions that follow it. The issue was also selected for the expected range of opinions to be found among in an extended community of MIT undergraduates, who were surveyed to produce substitute OMCS data, as discussed in the next section.

The document was manually divided into 34 sentences ($\{s_1 \ldots s_{34}\}$), and these sentences were manually decomposed into 49 assertions ($A = \{a_1 \ldots a_{49}\}$). These assertions included statements of simple phrasing, like "A fetus is innocent" and "A fetus is human." Each sentence was non-trivially associated with as many as 8 of these assertions. In all, there were a total of 93 non-zero values in the matrix R for this document.

7.2.2 Substitute for OMCS data

To compensate for the lack of controversy in the OMCS data set while simultaneously relieving the structure of the assertions from the need to fit the semantic network structure of ConceptNet, a survey was conducted to collect the opinions of volunteer contributors with respect to the 49 assertions extracted from the document.

98 contributors in the extended community of MIT undergraduates and graduate students responded to the survey. These contributors gave a total of 4323 ratings (where a rating is a contributor's opinion of an assertion), of which only 829 were neutral. This resulted in a matrix density of nearly 73% when preparing M_R, which should lead to a very strong PerspectiveSpace.

Collection interface

Figure 7-2 shows a representative portion of the web-based interface used to collect ratings from contributors. All 49 assertions are presented to each contributor in a random order chosen at the beginning of each page load. Table 7.1 explains the semantics of the rating choices presented to the contributors.

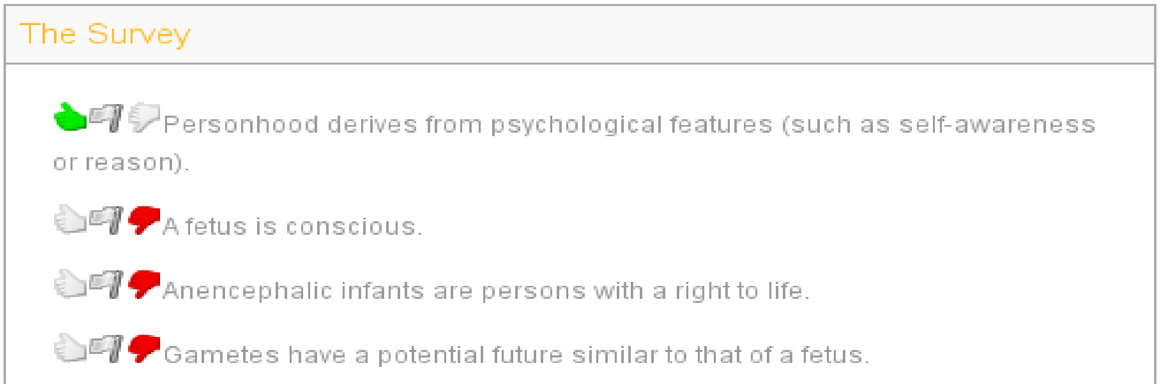


Figure 7-2: A screenshot of the SlantExplorer collection interface

Choice	Meaning	$M_R[i,j]$
None	No choice made	0
Thumbs-up	Agree with assertion	1.0
Thumbs-down	Disagree with assertion	-1.0
Flag	Contributor does not desire to express an opinion	0

Table 7.1: Semantics of the construction of PerspectiveSpace over the SlantExplorer data set

Analysis results

The data collected from the contributors were compiled to prepare a PerspectiveSpace for use with SlantExplorer. The first three axes of this PerspectiveSpace are characterized in tables 7.2, 7.3, and 7.4. These tables use the representation that would be used for subculture detection (see section 4.1), and they reveal a very successful preparation of a PerspectiveSpace. They show, for each of the three axes, the 15 most prominently placed assertions (positively or negatively) on each of these axes. As such, these tables can be taken to characterize the axes in question. They are arranged with the most positive placements at the top and the most negative placements at the bottom, with a line in the middle where the origin would fall on this axis—this line can also be taken to conceal the place where all other assertions in the data set would fall.

Table 7.2 shows the most dominant axis in the data set. It is clear from inspection that the positive[1] end of this axis is consistent with a pro-life mentality in the abortion

[1]It must be observed that the sign associated with the placement of an assertion on an axis makes no statement about the truth of the assertion or about the opinions of this author.

issue. Similarly, the negative end of the axis is consistent with the directly opposed pro-choice mentality in the abortion issue.

Placement	Assertion
0.188168721627	Contraception is wrong.
0.174843362685	A fetus is rational.
0.163473239525	A fetus is autonomous.
0.156694669739	Abortion is morally wrong because a fetus is an innocent human being.
0.130156804265	A fetus is the same entity as the adult into which it will develop.
0.130039026315	Abortion is wrong because it deprives a fetus of a valuable future.
0.127320113719	A fetus is self-conscious.
0.126072269748	A fetus is a person.
0.124238479189	A fetus is a person with rights.
-0.136014860912	A woman's right to control her own body outweighs a fetus' right to life.
-0.149099962593	Some killings are more wrong than others.
-0.160792451693	A reversibly comatose patient is a person.
-0.161862605481	No person has the right to use another's body against that person's will.
-0.1738071007	Killing an adult human being deprives the human of a future.
-0.174339648163	A woman has a right to control her own body.

Table 7.2: Characterization of the first principle perspective in the SlantExplorer data set

Table 7.3 shows that PerspectiveSpace has captured a much more subtle issue that can be explored with some independence from the much larger abortion issue. Inspection of this axis shows a concern for the issue of defining personhood. The positive end of the axis is generally consistent with the opinion that a human should have a few additional properties (like the ability to think) before being considered a person. The negative end of this axis is consistent with extending personhood to anything remotely human.

Table 7.4 is an axis that is defined entirely on the negative end of the axis. Inspection shows that agreement with the negative end of the axis is consistent with believing that personhood requires more than being human yet abortion is still objectionable.

Placement	Assertion
0.144269058929	Forcing a woman to continue an unwanted pregnancy is like forcing a person's body to be used as a dialysis machine for another person suffering from kidney failure.
0.14258033209	Rights derive from psychological features (such as self-awareness or reason).
0.14227338398	A human should be self-motivated in order to be a person.
0.141745247056	There is a difference between being a human and being a person.
0.13416894913	A human should be able to reason in order to be a person.
0.129192111147	Some people have a far stronger right to life than others.
0.128693568845	A human should be conscious (at least the capacity to feel pain) in order to be a person.
-0.123373328621	A fetus is a person.
-0.125068000403	Anencephalic infants are persons with a right to life.
-0.126363133172	It is wrong to deprive a person of a valuable future.
-0.129031627894	A woman, when pregnant from voluntary intercourse, has tacitly consented to the fetus using her body.
-0.140496706384	A fetus is a person with rights.
-0.140902017165	A fetus is human.
-0.145701091397	A person has the right to life, beginning at conception or whenever they come into existence.
-0.175669446205	All humans are persons.

Table 7.3: Characterization of the second principle perspective in the SlantExplorer data set

Placement	Assertion
-0.11535693517	Abortion is morally wrong because a fetus is an innocent human being.
-0.120857012243	Contraception deprives gametes of a potential future.
-0.122332482932	A fetus has a valuable future.
-0.138256337806	Personhood derives from psychological features (such as self-awareness or reason).
-0.145519666606	A human should be conscious (at least the capacity to feel pain) in order to be a person.
-0.147322085166	A fetus is innocent.
-0.149718118129	Some people's futures are more valuable than others.
-0.149737414799	Some people have a far stronger right to life than others.
-0.150533064452	There is a difference between being a human and being a person.
-0.15364831398	Abortion is wrong because it deprives a fetus of a valuable future.
-0.159099417053	The killing of a human being may be morally justified if that human has no valuable future.
-0.159250094567	Abortion deprives a fetus of a valuable future.
-0.163240247771	Some killings are more wrong than others.
-0.174507356353	Rights derive from psychological features (such as self-awareness or reason).
-0.210567981614	Abortion can be justified when the same justification could be applied to killing an adult human.

Table 7.4: Characterization of the third principle perspective in the SlantExplorer data set

Inspection of tables 7.2, 7.3, and 7.4 should also readily reveal a steadily declining coherence. The first axis was clearly defined along lines of pro-life vs. pro-choice. The second axis was fairly clearly defined along lines of loose and strict interpretations of personhood, but there was a stray placement of "Forcing a woman to continue an unwanted pregnancy is like forcing a person's body to be used as a dialysis machine..." that is not quite consistent with that definition. The third axis, though clearly touching on some right-to-life issues not covered in previous axes, has features otherwise found in the second axis. This degradation across axes is an expected phenomenon when using SVD in any analysis.

The analyses of these axes shows that the data set used for SlantExplorer in lieu of OMCS is valid. Furthermore, this reinforces the claim in section 4.1 that applications can be built around studying these axes by inspection (as done here), using PerspectiveSpace as a means for detecting subcultures.

7.2.3 Interaction design

Of the possible annotation modes conceived for SlantExplorer, only one was implemented in the prototype: revealing dominant perspectives out of a select subset, as described in section 7.1.4. Here, the select subset was the set of the three most dominant axes in the SlantExplorer PerspectiveSpace.

Figure 7-3 shows a representative screenshot of the web-based interface for viewing a SlantExplorer annotation report. The three major components are the summary, the legend, and the annotated document.

The three most descriptive axes (k_1, k_2, and k_3) are identified in the summary box on the left of the screen. Here, it is revealed that axes 0, 1, and 2 (which correspond to the top three principle perspectives of the underlying data set) are also the most descriptive axes for the document. As the data set was prepared for the document presented, this is to be expected, but documents in the ideal SlantExplorer design are much more likely to diverge from the dominant trends of the OMCS data set. The percentages shown are indications of how much of a document's total activity (the sum of all q_k as defined in equation 7.3) is given by the activity in that particular axis (q_{k_1}, q_{k_2}, and q_{k_3}).

The legend, which appears under the summary box, again identifies the axes k_1, k_2, and k_3, but here each end of each axis is associated with a primary color. In this example, the legend shows that text best associated with the positive end of axis 0 (k_1) will be written in green type, and the text best associated with the negative end of axis 0 will be written in red type. Other primary and secondary colors were chosen for axes k_2 and k_3 so as to maximize contrast with each other and maximize the legibility of the document. Text that received no annotation at all is written in a neutral gray.

7.3 Future directions

The full realization of the SlantExplorer concept is an obvious future direction to be taken once natural language processing technologies and common sense knowledge resources like OMCS develop more fully. This prototype does, however, leave open interesting questions to be explored.

In particular, if the OMCS data set were to be more fully developed, would axes found in this much more restrictive data set still be as coherent, even if they were relegated to the ranks of 32$^{\text{nd}}$ most-significant axis and below? That is, would the presence of higher-ranking axes interfere with the clarity that might be need to conduct subculture detection through inspection? If this is the case, then a possible solution would be to borrow the idea of realm filtering first introduced in one of the earliest manifestations of the ConceptNet project [22].

Realm filtering was a way to improve the quality of the ConceptNet data set in a particular application by restricting the portion of ConceptNet used to within a limited semantic distance of concepts belonging to the application. A similar principle can be applied to the preparation of a PerspectiveSpace for use in SlantExplorer: the M_R matrix used to prepare PerspectiveSpace can be weighted such that the assertions most relevant to a document being analyzed have a greater magnitude than assertions that are not. With this alteration, the assertions having to do with a document being

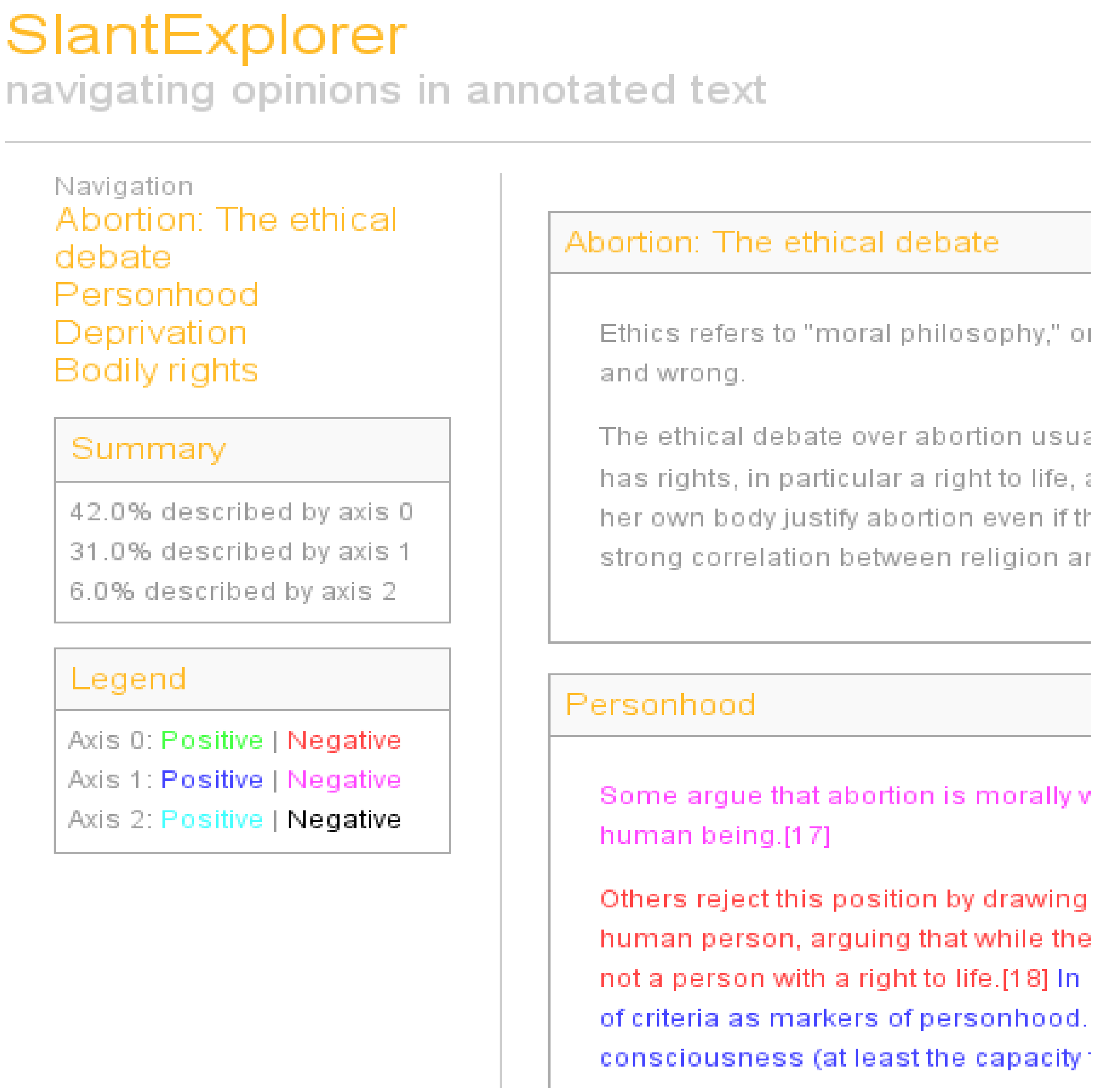


Figure 7-3: A screenshot of the SlantExplorer analysis interface

analyzed should define higher-ranking principle perspectives, but the more weakly-weighted assertions will still help shape PerspectiveSpace according to latent belief patterns.

Chapter 8

2-wit

2-wit, whose name is a contraction of "What Will I Think," is a web-based system that collects opinions people have about movies and suggests, to each user, opinions from other users that he or she might agree with.

2-wit was implemented as an interactive web site built on the Django [14] web development framework and the Divisi [15] SVD abstraction toolkit developed at the MIT Media Laboratory by a team including this author. The principles of PerspectiveSpace drive 2-wit's underlying rating system and data collection for 2-wit.

The architecture of 2-wit was designed to support reviews of more than just movies, provided that an implementation of an interface for users to interact with each class of reviewable item was provided. So far, however, only a movie interface was implemented.

As the 2-wit interface (see figure 3-1), the matrix-population semantics (see table 3.1), and the nuances of preparing PerspectiveSpace from the 2-wit data set are all covered in chapter 3 as an extended example, these details are not covered here.

8.1 Data model

Before discussing the particulars and results of each application of 2-wit, it is important to establish the terminology used in the implementation. In 2-wit, there are six core data structures, which are illustrated in figure 8-1 and defined as follows:

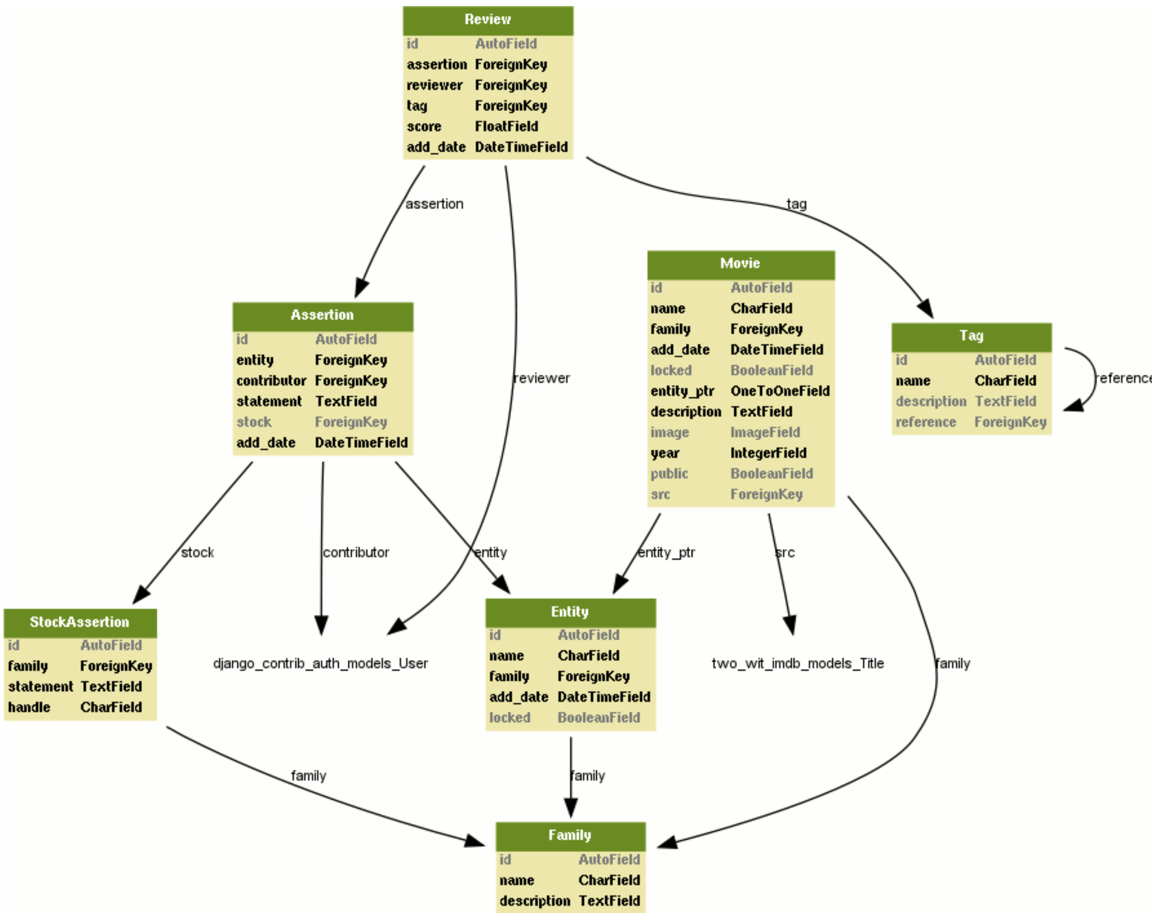


Figure 8-1: Model diagram of 2-wit data structures

User A user is a person that can contribute opinions and for whom applicable opinions are selected. Each user-tag pair[1] with a non-trivial number of contributions is assigned a row of the matrix for the computation of PerspectiveSpace.

Family A family is a category of entities. "Movies" and "restaurants" are two examples of families.

Entity An entity is an item belonging to a family. A particular movie or a particular restaurant could be an entity, for example. Entities are present to organize opinions for interface purposes, as they play no mathematically significant role in the computation of PerspectiveSpace. They do play a role, however, in the leave-one-out analysis for the evaluation of 2-wit, which is explained in section 8.4.

Assertion An assertion is a free-text statement made about a particular entity. For quality control purposes, each assertion record includes a reference to the contributor that entered the assertion as well as a timestamp for the date of entry.

Tag A tag is a special modifier that extends assertions. The two tags used by 2-wit so far have equivalent semantics to the expressions "this user agrees with this assertion" (the "agree" tag) and "This user considers this assertion to be spam, obscene, or similarly inappropriate" (the "junk" tag). Each combination of a tag and a user comprises a row of M_R for constructing PerspectiveSpace.

Review A review is the combination of a user, assertion, tag, and a boolean value that indicates the user's opinion of attaching the tag to the assertion, and as such, a Review represents an entry in the M_R matrix. The exact value used in the matrix was subject to fine-tuning, but the values are floating-point numbers of either positive or negative sign. Each user is presumed to agree with each statement he or she contributes, resulting in a positive value with the assertion and the "agree" tag, and

[1] Note that 2-wit uses the tagged model of matrix construction.

each user is presumed to consider each assertion her or she contributes as not spam, resulting in a negative value with the assertion and the “junk” tag.

8.2 Thinkerprints

The perspective vector for each user is exposed to the user as a fun way for the user to see some of the information that goes into making recommendations presented to the user—this information takes the form of a “Thinkerprint,” a short alphabetical string in the spirit of psychometric indicators like the Myers-Briggs Type Indicator [9]. An example Thinkerprint is shown in figure 8-2.

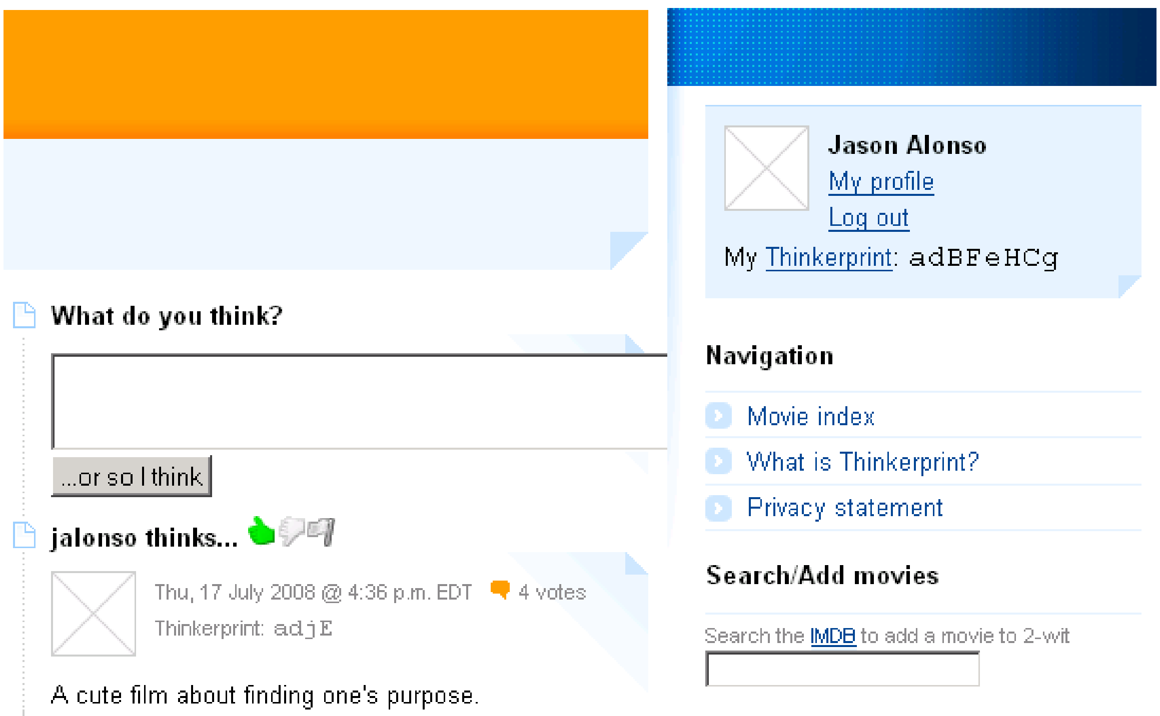


Figure 8-2: Screenshot of 2-wit’s Thinkerprint display

The algorithm for computing a Thinkerprint can be seen in appendix A.1 on page 100. A summary follows here:

1. Load the perspective vector, weighted by Σ for the relevant person or assertion.

2. Examine the top 26 axes (one for each letter of the alphabet) only, and assign

these axes the letters of the alphabet in order, with the most prominent axis being assigned the letter A.

3. Sort these axes according to the magnitude of the perspective vector along each axis.

4. Change the capitalization of each letter to reflect whether or not the perspective vector falls in the positive end (for an uppercase letter) or the negative end (for a lowercase letter).

5. Keep only the top 8 axes for a person or the top 4 axes for an assertion.

8.3 Movie reviews and ratings

The 2-wit implementation that covers movie entities was designed to give people a social space where they could share their opinions on the movies they've seen, with the general expectation that they could also rely on each other to help make decisions on movies to see in the future. Though the interface was designed to emphasize the social nature of the system, it is not a social networking system that allows people to declare "friends" or "groups" as contemporary social networking sites like Facebook permit.

The movie review domain was chosen as for any two people in the same society, there are likely to be at least a few movies that they have both seen. At the same time, people's tastes and opinions of movies are sufficiently varied to make a range of trends readily detectable. As such, even a small number of contributors was expected to produce usable results.

At the time of this writing, a total of 26 users contributed a total of 51 assertions on 14 movies, assigned 188 reviews on those assertions with the "agree" tag (meaning that they expressed an opinion of either agreement or disagreement), and assigned 200 reviews with the "junk" tag (meaning they declared whether or not they believed the assertions to be junk).

8.4 Evaluation

A variation of the leave-one-out test was performed to test the interpolative abilities of PerspectiveSpace in the 2-wit implementation. Since the presumed usage scenario for the 2-wit movie system is that a user would search for believable reviews of movies that he or she has not seen, the traditional leave-one-out test was extended to leave out one user-entity pair per trial. As such, all reviews a particular user gave for assertions about a particular entity were removed for the duration of a trial, and recall was defined in terms of the number of reviews that could be properly estimated. The test, accordingly, is dubbed "leave-some-out."

8.4.1 Degenerate cases

In determining viable user-entity pairs for the leave-some-out test, the test skipped the degenerate case of assertions where only one user provided a review was omitted in trials testing the applicable user and the relevant entity. These omissions were consolidated to skip the degenerate case of entities where only one user contributed reviews. Finally, the case of users who commented on only one entity was also avoided, as there were no grounds for an SVD to interpolate reviews for such users. The precise mechanism for avoiding degenerate cases can be examined in the source listing for the procedure in appendix A.2 on page 102.

Given these restrictions, it was determined that exactly 61 user-entity pairs were viable as trials for the evaluation, and so the evaluation covers all possible user-entity pairs rather than a random subset. This test was also repeated with varying values of k, which determines the number of axes that should be computed and used for the analysis.

8.4.2 Results

The results are shown in table 8.1. For the purposes of this evaluation, "hits" are defined as the number of trials in which the review of an assertion was successfully interpolated using perspective projection[2] with $\tau = 0.01$. "Misses" are defined similarly where the review of an assertion was predicted opposite to the the correct value. "Undecided" is defined as the number of assertions for which the magnitude of the predictive rating was smaller than τ, which is representative of the cases where there is insufficient confidence to make a reasonable estimate of a user's acceptance or rejection of an assertion. The "% decided" represents the fraction of assertions for which a confident estimation was made, while "% correct" represents the portion of *the decided assertions* for which the correct estimation was made.

8.4.3 Conclusions

The results are positive in that 2-wit performed substantially better than chance in estimating the opinions that users would have about movies. Given any user-movie pair for testing, the leave-some-out test appropriately excluded all reviews the user made of any assertion about the movie. It is further interesting to note that the best performance was obtained with $k = 4$—a reasonable interpretation of this phenomenon is that the omission of lower-ranking axes removed noise from the data set, particularly excluding "information" that specifically worked to diminish properly-interpolated ratings for which there was no direct measurement.

		Cumulative					Average	
Tags	k	Hits	Misses	Undecided	% decided	% correct	% decided	% correct
all	4	215	74	88	77	74	76	78
	8	215	74	88	77	74	78	74
	16	150	81	146	61	65	63	66
agree	4	93	46	43	76	67	76	71
	8	92	42	48	74	69	75	69
	16	58	45	79	57	56	60	59
junk	4	122	28	44	77	81	76	84
	8	123	32	39	80	79	81	79
	16	92	36	66	66	72	65	71

Table 8.1: Results of the 2-wit evaluation

[2] See section 4.4 for a discussion of perspective projection.

8.5 Future directions

2-wit, as a web-based service in a Web 2.0 world, has a plethora of future directions in which it can go. This involves such things as the development of miniature 2-wit applications that can be linked into a traditional social networking site, as this would increase the system's exposure to contributors as well as increase the user base. Widgets can be developed to let bloggers and other website maintainers embed a review interface from 2-wit into their own pages in reference to a movie.

These directions are most compelling, however, when considered with a few more academic advances:

8.5.1 Restaurant reviews

Restaurants are a natural addition to 2-wit, given the popular "dinner and movie" combination often used for planning a date.

PerspectiveSpace could help prospective restaurant patrons understand otherwise vaguely-defined properties of a restaurant, like its atmosphere, which can draw different descriptions from different branches of an urban community. A graduate student and a financial adviser, for example, can very easily have different interpretations of "dressy attire."

Since the role PerspectiveSpace can play in restaurant review can be extremely social in nature, it must be observed that PerspectiveSpace can essentially serve as a tool for helping people socially navigate and understand their surrounding communities.

8.5.2 General product reviews

Another interesting direction for 2-wit is to support reviews of consumer products, allowing people to hear opinions on things from like-minded people. This application domain becomes extremely compelling when coupled with the idea of building a review widget that website maintainers can embed into their own pages. In this scenario, multiple vendors of the same product can rely on the same pool of reviews, and customers new to a site can place better trust in the opinions displayed.

8.5.3 Cross-family analysis

Once more than one family of entities becomes available to 2-wit, it would be interesting to see how 2-wit's performance is affected by including multiple families in a PerspectiveSpace computation. Two people with similar opinions on movies may have similar opinions on restaurants, for example.

The data model used by 2-wit naturally accommodates testing this hypothesis, since it readily permits the construction of an $M_{R_{MOVIES}}$ matrix for consolidating movie opinions separately from an $M_{R_{RESTAURANTS}}$ matrix for consolidating restaurant opinions.

Appendix A

2-wit source code

This appendix is dedicated to holding samples from the source code of 2-wit, current as of revision 296. It is by no means a complete representation of all code developed by this author to make this system operational. Only those elements directly referenced in the course of this paper are included here.

At the time of this writing, 2-wit is operational at `http://2-wit.media.mit.edu/` and is expected to remain operational for the indefinite future. The project itself should be archived at the site of the Commonsense Computing Initiative at `http://csc.media.mit.edu/`.

A.1 reviews/thinkerprint.py

```
from reviews.processes import load_tensor_set


def _thinkercode(axis, value):
    from string import uppercase, lowercase
    if value < 0: return lowercase[axis]
    else: return uppercase[axis]


def _scale_vector(vec, svd):
    d = svd.core.shape[0]
    res = {}
    for axis in range(d):
        res[axis] = vec[(axis,)] * svd.core[(axis,axis)]
    return res


def user_print(user, user_tag, family_name='', tag_name='', max_length=8):
    # Get svd
    tensor, norm, svd = load_tensor_set( family_name, tag_name )
```

```
    # Look for the user's record
    if (user.username, user_tag) not in svd.u.label_lists()[0]: return None
    vec = _scale_vector(svd.u[(user.username, user_tag),:], svd)

    # Sort the results by axial significance
    axes = sorted(vec.items(), key=lambda x: (abs(x[1]),-x[0]), reverse=True)
    axes = [(k,v) for (k,v) in axes if k < 26]
    codes = [_thinkercode(k,v) for (k,v) in axes]

    # Operation Complete!
    return ''.join(codes[:max_length])

def assertion_print(assertion, family_name='', tag_name='', max_length=8):
    # Get svd
    tensor, norm, svd = load_tensor_set( family_name, tag_name )

    # Look for the assertion's record
    if assertion.id not in svd.v.label_lists()[0]: return None
    vec = _scale_vector(svd.v[assertion.id,:], svd)
```

```
    # Sort the results by axial significance
    axes = sorted(vec.items(), key=lambda x: (abs(x[1]),-x[0]), reverse=True)
    axes = [(k,v) for (k,v) in axes if k < 26]
    codes = [_thinkercode(k,v) for (k,v) in axes]

    # Operation Complete!
    return ''.join(codes[:max_length])
```

A.2 evaluation.py


```
from reviews.models import Family, Tag, Review
from reviews.processes import build_tensor, normalize_tensor

def find_valid_tests(family, tags=None):
    # Handle default arguments
    if tags is None:
        tags = Tag.objects.all()

    # Accumulate reviewers of assertions from the set of applicable reviews
    asrt_reviewers = {}
```

```
for review in Review.objects.filter(
                assertion__entity__family=family,
                tag__in=[tag.id for tag in tags],
                ):
        revr_set = asrt_reviewers.setdefault(review.assertion,set())
        revr_set.add(review.reviewer)

# Find assertions that are not degenerate
asrt_testable = [       asn for asn
                        in asrt_reviewers
                        if len(asrt_reviewers[asn]) > 1
                ]

# Accumulate entities from non-degenerate assertions
entity_reviewers = {}
for asn in asrt_testable:
        revr_set = entity_reviewers.setdefault(asn.entity,set())
        revr_set.update(asrt_reviewers[asn])

# Find entities that are not degenerate
```

```
entities_testable = [   entity for entity
                        in entity_reviewers
                        if len(entity_reviewers[entity]) > 1
                      ]
# Find users who gave non-degenerate reviews of non-degenerate entities
revr_entities = {}
for entity in entities_testable:
        for revr in entity_reviewers[entity]:
                entity_set = revr_entities.setdefault(revr,set())
                entity_set.add(entity)

# Find users who are not degenerate
reviewers_testable = [ revr for revr
                        in revr_entities
                        if len(revr_entities[revr]) > 1
                      ]

# Accumulate valid tests and their test values
for revr in reviewers_testable:
        for entity in revr_entities[revr]:
```

```
        # Find all reviews applicable to a test
        test_values = {}
        for review in Review.objects.filter(
                assertion__entity=entity,
                tag__in=[tag.id for tag in tags],
                reviewer=revr,
                ):
            # Make sure the test case is not degenerate
            if review.assertion not in asrt_testable:
                continue

            # Record the test values
            row = (revr.username, review.tag.name)
            col = review.assertion.id
            test_values[row,col] = review.score

        yield family, revr, entity, test_values

# Operation Complete!
```

```
def run_test(family, user, entity, test_values, k=8, tau=0.01, tag=None):
    # Build a tensor for testing
    tensor = build_tensor(family, tag, ignore=[(user, entity)])

    # Normalize the tensor
    norm = normalize_tensor(tensor)

    # Decompose the normalized tensor
    svd = norm.svd(k=k)

    # Classify each test result
    hits = 0
    misses = 0
    undecided = 0
    for coord, score_trg in test_values.items():
        score_est = svd.get_ahat(coord)
        if abs(score_est) < tau:
            undecided += 1
        elif (score_est > 0.0) == (score_trg > 0.0):
            hits += 1
```

```
        else:
            misses += 1

    # Operation Complete!
    return hits, misses, undecided

def generate_test_results(family, tag=None, **kwargs):
    # Handle default arguments
    if tag is None:
        tags = Tag.objects.all()
    else:
        tags = [tag]

    # Generate valid tests
    test_gen = find_valid_tests(family, tags)

    # Generate test results
    for test in test_gen:
        result = run_test(*test, **kwargs)
        yield result
```

```
#if __name__ == '__main__':
if True:
    # Obtain the Movies family
    family = Family.objects.get(name='Movies')

    # Obtain the agree and junk tags
    tag_agree = Tag.objects.get(name='agree')
    tag_junk = Tag.objects.get(name='junk')

    # Set k
    k = 8

    # Run the test on each of the possible tags
    for tag in [None, tag_agree, tag_junk]:
        # Open the result file
        if tag is None: fn = 'all-k%d.csv' % k
        else: fn = '%s-k%d.csv' % (tag.name, k)
        result_file = open(fn, 'w')
```

```
# Record test results
for result in generate_test_results(family,tag,k=k):
    line = "%d,%d,%d\n" % result
    result_file.write(line)
    result_file.flush()

# Close the result file
result_file.close()
```

Bibliography

[1] Jason B. Alonso. CSAMOA: A common sense application model of architecture. *Proceedings of the Workshop on Common Sense and Interactive Applications*, 2007.

[2] Junia Anacleto, Henry Lieberman, Marie Tsutsumi, Vania Neris, Aparecido Carvalho, Jose Espinosa, and Silvia Zem-Mascarenhas. Can common sense uncover cultural differences in computer applications? In *Proceedings of IFIP World Computer Conference*, Santiago, Chile, 2006.

[3] Junia Coutinho Anacleto, Aparecido Fabiano Pinatti de Carvalho, Vânia Paula de Almeida Néris, Muriel de Souza Godoi, Silva Zem-Mascarenhas, and Américo Talarico Neto. In *Proceedings of XVII Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 217–226, 2006.

[4] Kenneth Arnold, Jason Alonso, Catherine Havasi, Robert Speer, and Henry Lieberman. Conceptnet. Web site, 2008. `http://conceptnet.media.mit.edu/`; accessed 13-August-2008.

[5] David J. Bartholomew. *Measuring Intelligence: Facts and Fallacies*. Cambridge University Press, 2004.

[6] Carnegie Mellon University. reCAPTCHA: Stop Spam, Read Books. reCAPTCHA project web site, 2008. `http://recaptcha.net/`.

[7] Aparecido Fabiano Pinatti de Carvalho, Junia Coutinho Anacleto, Henry Lieberman, Muriel de Souza Godoi, and Silvia Zem-Mascarenhas. Using common sense for planning learning activities. *In Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, 2007.

[8] Ian Scott Eslick. Searching for commonsense. Master's thesis, MIT Media Laboratory, 2006.

[9] Myers & Briggs Foundation. My MBTI Personality Type - MBTI Basics. Foundation web site, 2008. `http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/`; accessed 18-August-2008.

[10] Python Software Foundation. Python programming language. Python web site, 2006. `http://www.python.org`.

[11] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992. http://doi.acm.org/10.1145/138859.138867.

[12] Catherine Havasi, James Pustejovsky, and Anna Rumshisky. An evaluation of the brandeis semantic ontology. *Proceedings of the Fourth International Workshop on Generative Approaches to the Lexicon*, 2007.

[13] Catherine Havasi, Robert Speer, and Jason Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, 2007.

[14] Lawrence Journal-World. Django, the web framework for perfectionists with deadlines. Django web site, 2006. http://www.djangoproject.com.

[15] Kenneth Arnold and Robert Speer and Catherine Havasi and Jason Alonso and Jayant Krishnamurthy. Divisi: Commonsense Reasoning over Semantic Networks. Project web site, 2008. http://divisi.media.mit.edu/; accessed 18-August-2008.

[16] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. He says, she says: conflict and coordination in wikipedia. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 453–462, New York, NY, USA, 2007. ACM. http://doi.acm.org/10.1145/1240624.1240698.

[17] Cliff Lampe and Paul Resnick. Slash(dot) and burn: distributed moderation in a large online conversation space. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 543–550, New York, NY, USA, 2004. ACM. http://doi.acm.org/10.1145/985692.985761.

[18] Cliff A.C. Lampe, Erik Johnston, and Paul Resnick. Follow the reader: filtering comments on slashdot. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1253–1262, New York, NY, USA, 2007. ACM. http://doi.acm.org/10.1145/1240624.1240815.

[19] Doug Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 11:33–38, 1995.

[20] Henry Lieberman and Ashwani Kumar. Providing expert advice by analogy for on-line help. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 26–32, Washington, DC, USA, 2005. IEEE Computer Society. http://dx.doi.org/10.1109/WI.2005.121.

[21] Hugo Liu and Pattie Maes. What would they think?: a computational model of attitudes. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 38–45, New York, NY, USA, 2004. ACM. http://doi.acm.org/10.1145/964442.964451.

[22] Hugo Liu and Push Singh. Conceptnet: a practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4), 2004. Kluwer Academic Publishers.

[23] Reid Priedhorsky, Jilin Chen, Shyong (Tony) K. Lam, Katherine Panciera, Loren Terveen, and John Riedl. Creating, destroying, and restoring value in wikipedia. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268, New York, NY, USA, 2007. ACM. http://doi.acm.org/10.1145/1316624.1316663.

[24] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM. http://doi.acm.org/10.1145/192844.192905.

[25] Push Singh. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. AAAI, 2002.

[26] SourceForge, Inc. Slashdot: News for nerds, stuff that matters. Slashdot web site, 2008. http://slashdot.org/.

[27] Robert Speer. Learning common sense knowledge from user interaction and principal component analysis. M.Eng. thesis, MIT Media Lab, 2006. http://web.media.mit.edu/~lieber/Teaching/Common-Sense-Course/Speer-Thesis.pdf.

[28] Robert Speer, Catherine Havasi, Jason Alonso, Kenneth Arnold, and Henry Lieberman. Analogyspace. Web site, 2007. http://analogyspace.media.mit.edu/; accessed 13-August-2008.

[29] Robert Speer, Catherine Havasi, Jason Alonso, Kenneth Arnold, and Henry Lieberman. Add new knowledge - Open Mind Common Sense. OMCS web site, 2008. http://commons.media.mit.edu/en/add/.

[30] Robert Speer, Catherine Havasi, and Henry Lieberman. AnalogySpace: Reducing the dimensionality of common sense knowledge. *Proceedings of AAAI 2008*, July 2008.

[31] Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):56–60, 2004. http://doi.acm.org/10.1145/966389.966390.

[32] Ba-Quy Vuong, Ee-Peng Lim, Aixin Sun, Minh-Tam Le, and Hady Wirawan Lauw. On ranking controversies in wikipedia: models and evaluation. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 171–182, New York, NY, USA, 2008. ACM. http://doi.acm.org/10.1145/1341531.1341556.

[33] Wikimedia Foundation, Inc. Ethical aspects of abortion — Wikipedia, The Free Encyclopedia. English language Wikipedia web site, 2008. `http://en.wikipedia.org/w/index.php?title=Ethical_aspects_of_abortion&oldid=197476053`; accessed 25-March-2008.

[34] Wikimedia Foundation, Inc. Wikipedia, the free encyclopedia. English language Wikipedia web site, 2008. `http://en.wikipedia.org/`.