SPORK: A SUMMARIZATION PIPELINE FOR ONLINE REPOSITORIES OF KNOWLEDGE

A Thesis

presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Steffen Lyngbaek

June 2013

COMMITTEE MEMBERSHIP

TITLE:	SPORK: A Summarization Pipeline for Online Repositories of Knowledge
AUTHOR:	Steffen Lyngbaek
DATE SUBMITTED:	June 2013
COMMITTEE CHAIR:	Professor Alexander Dekhtyar, Ph.D., Deparment of Computer Science
COMMITTEE MEMBER:	Professor Franz Kurfess, Ph.D., Deparment of Computer Science
COMMITTEE MEMBER:	Professor Foaad Khosmood, Ph.D., Deparment of Computer Science

Abstract

SPORK: A Summarization Pipeline for Online Repositories of Knowledge

Steffen Lyngbaek

The web 2.0 era has ushered an unprecedented amount of interactivity on the Internet resulting in a flood of user-generated content. This content is often unstructured and comes in the form of blog posts and comment discussions. Users can no longer keep up with the amount of content available, which causes developers to start relying on natural language techniques to help mitigate the problem. Although many natural language processing techniques have been employed for years, automatic text summarization, in particular, has recently gained traction. This research proposes a graph-based, extractive text summarization system called SPORK (Summarization Pipeline for Online Repositories of Knowledge).

The goal of SPORK is to be able to identify important key topics presented in multi-document texts, such as online comment threads. While most other automatic summarization systems simply focus on finding the top sentences represented in the text, SPORK separates the text into clusters, and identifies different topics and opinions presented in the text. SPORK has shown results of managing to identify 72% of key topics present in any discussion and up to 80% of key topics in a well-structured discussion.

Acknowledgements

Thanks to:

- **My Advisor:** Alexander Dekhtyar
- **My Panel of Experts:** Aldrin Montana, Alex Boese, Brett Armstrong, Eriq Augustine, Evan Gray, Giovanni Prinzivalli, James Neumann, Joseph White, Mike Buerli, Taylor Graybehl, Tyler Harper

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As websites on the Internet in the Web 2.0 era have become more interactive, there has been an explosion of new user-generated content. Sites like Wikipedia have surged to around 4 million articles just in English generated completely by the community [7]. Various blogs and forums now get content generated by the audience in the form of comments and posts. Additionally, social networking services like Twitter and Facebook have created an outlet for people to discuss, comment, rant, and simply socialize. While the amount of content users consume has grown drastically, the way it is consumed has not evolved. This causes the problem of information overload where users can miss out on relevant information.

Natural language processing has been used extensively on the Internet and has been put to a variety of tasks to mitigate this problem. Some of the solutions include word clouds, trending topics, and aggregated reviews for sites, such as Yelp and Amazon. Keyword extraction can be used to better tag certain documents, like research articles. Additionally, sites like Amazon and Netflix have created recommendation engines, all based in part on natural language process-

ing, which help users sift through the enormous amount of content to help find relevant matches. Search engines rely on the use of natural language processing to return more relevant results. They use various techniques for information retrieval and for getting a deeper semantic understanding of what search queries mean. Text extraction and summarization can be used to display summaries of search results and is widely used in question and answering systems.

While many natural language processing techniques can benefit users, automatic text summarization is the key to mitigating information overload produced by these sources. According to [36], "a summary can be loosely defined as a text that is produced from one or more text(s), that conveys important information in the original text(s), and that is no longer than half of the original text and usually significantly less." The idea is that automatic summarization should reduce the amount of text in a document or multiple documents while still retaining all of the important information. There are several different types of text summarization all suited to solve slightly different problems. A detailed analysis of current summarization techniques and a novel approach will be explained in later chapters.

A perfect example of a service with community-driven content that is well suited for automatic summarization is Reddit [4]. Reddit is a community-operated site that lets users upload news stories found anywhere on the Internet. Other users can then vote and comment on these different submitted news stories. Reddit is currently the 54^{th} most popular website in the United States according to Alexa Rank and presents an overload of information to web users in the form of posts and comments [5]. Several popular posts contain upwards of a thousand comments, which makes it unrealistic for users to extract all the relevant

information needed in the post. In addition, posts are ranked by the number of votes they get, and use a hierarchical structure based on parent and response comments.

I plan to address the problem of information overload, specifically within streaming and community driven content, such as comment threads from web blogs and forum posts. Although Reddit is the source of content used for this work, the techniques explained in later chapters can be applied to other similar sources. The system created in this work is a novel attempt at trying to extract the "ground truth" out of a discussion that contains many different topics. We call this system SPORK (summarization pipeline for online repositories of knowledge) and it is aimed at not only segmenting the various opinions scattered throughout a text, but also at ranking and finding the most relevant and important ones. SPORK employs a plethora of natural language processing techniques to create a summarization pipeline that under best circumstances is able to identify up to 80% of the important topics within the text. This is a significant achievement and highlights the contributions of our work.

Although SPORK is accessed through a CLI (command line interface), future work can extend it into a browser plugin that can be activated for various sites. As stated before, the ultimate end goal of this system is to mitigate information overload. A browser plugin that automatically summarizes discussions online and displays the important results above the comment threads help solve this problem. This plugin would be backed by the sound results of the SPORK pipeline. This work focuses solely on the natural language processing back end and a plugin that wraps the SPORK pipeline is seen purely as future work.

The rest of this work is split up as such: Chapter 2 describes the background

knowledge required and various existing techniques for summarization. Chapter 3 reveals the design and implementation of the SPORK pipeline. Chapter 4 contains the data gathering and evaluation of the results. Finally, Chapter 5 concludes the paper with the a summary overview and potential future work.

Chapter 2

Background

Automatic text summarization is an active and developing field, and one of the important applications within natural language processing. Within automatic text summarization, several techniques are used to accomplish a variety of tasks. Generally speaking, there are two distinctive types of text summarization: extractive and abstractive. Extractive summarization approaches create a summary by using sentences or phrases from the corpus text. They focus on choosing relevant and salient sentences that can be used to represent the document. On the other hand, abstractive summarization approaches create a novel summary from a corpus text. This type of technique can rely on creating a semantic representation of the text and uses natural language generation to create novel sentences that do not necessarily show up in the corpus text. In general, most techniques that are not defined as an extractive method are still considered abstractive or partially abstractive approaches. Recently, abstractive summarization techniques have begun to see some research and development; however, a vast majority of the work in the field still relies on extractive approaches.

2.1 Extractive Summarization

While both extractive the and abstractive methods of summarization are explored and compared, this chapter begins with analyzing the extractive method because of its historical foundation and established development in the field. Additionally, many of the techniques discussed below are relevant to both abstractive and extractive approaches. Most current research has gone into extractive summarization including but not limited to: [20, 21, 28, 40, 16, 11, 32, 13], and [13]. These works use a variety of techniques, and can be broken up into several categories.

Identifying the differences in approaches to text summarization is only part of the solution. Extracting salient sentences is required for making an extractive summarization technique useful. There have been several intrinsic methods used, such as using sentence surface features [40] or simply using word frequencies as in [28]. One of the most popular methods used in extractive summarization takes a graph-based approach. These approaches have been shown in: [20, 21, 32, 13]. The TextRank and LexRank research are some of the most popular and most cited. This research predominantly revolves around the PageRank algorithm written by Larry Page. PageRank is a graph-based approach used for ranking the importance of websites. Using this algorithm, each website acts as a node in the graph, and its hyperlinks become the edges in the graph. The importance of a website is directly related to the number of other linking websites or edges and how important those linking pages are. According to PageRank, your site will have a higher rank if a comparable site of high importance links to your site rather than a site of low importance. This intuitive algorithm can be applied very well in the context of text summarization. In TextRank and LexRank, sentences of a

text act as the nodes of a graph while the edges between nodes act as a similarity measure between sentences. For example, edges can be represented as a simple cosine similarity or as a complex semantic similarity measure. Although these two methods work with sentences, any scope of text can act as a node whether it be a word, sentence, paragraph, comment, or even document. The similarity measures vary in TextRank and LexRank, but the idea behind them is the same.

The work done in [20] and [21] takes the PageRank algorithm a little further and incorporates three different aspects to the sentence extraction model. They introduce a word representativeness score, which is broken up into 3 separate components: reader measure, quotation measure, and topic measure. The reader measure uses a PageRank-like algorithm, which determines the reader authority and works on a blog level instead of a blog post level. The reader authority measure is used to calculate the value of a word in the blog post based on which authors used that word in their comments combined with their respective author authority. The quotation measure works on a blog post level and measures the value of a word in the post. This value reflects the comments in which the word appears, and how many times those respective comments have been quoted by other comments. Lastly, the topic measure uses a clustering algorithm to determine how relevant certain topics are, and how close a given comment is to the center of a one of the topic clusters. Not only does this technique somewhat involve topic segmentation, it also focuses on web comments and attempts to reduce noise in the text. The authors state that these three methods reduce susceptibility to noise and spam. The commenters and bloggers have no control over the reader measure and very little control over the quotation and topic measures.

Although these methods might produce salient sentences for summarization, they do little to understand the semantics behind the sentences. Therefore, they often do a poor job capturing all of the pertinent information in the document. This is not necessarily bad, but in multi-document summarization and in web comment threads, specifically, there can be multiple viewpoints and ideas expressed. If one or more sentences is produced by one of the techniques above, there will likely be a large amount of overlap between the sentences. Since the score of a sentence is based on sentence similarity, similar sentences will likely be ranked at the top. To solve this issue, some research like [13] uses studies from [10] to apply an approach known as Maximal Marginal Relevance, MMR. According to [10], MMR is a technique that is used to "reduce redundancy while maintaining query relevance in re-ranking retrieved documents and in selecting appropriate passages for text summarization." While this technique and others may reduce redundancy, it does not ensure a comprehensive summarization on all important aspects of a document.

2.2 Supervised Summarization

Automatic text summarization methods can use a supervised, unsupervised, or semi-supervised approach. A supervised or semi-supervised approach requires manually labeled data to train on. As noted in [25], getting labeled data can be very expensive and time consuming. In addition, when training on labeled data, all data comes from one domain. This means that running the summarization will only work for that specific domain, minimizing the portability of the approach. To mitigate the issue of getting expensive labeled data, research such as [40] has used a semi-supervised approach. This approach combines labeled and unlabeled

data, and manages to achieve similar results as a fully-supervised implementation. However, this does not eliminate the portability issue, and only sidesteps the issue of acquiring labeled data. In addition, research has not suggested that supervised approaches provide more accurate or better quality summaries. Therefore, these points and the much larger scope of research done on unsupervised approaches are why this work focuses on unsupervised text summarization.

2.3 Multi-Document Summarization

In addition to taking an unsupervised approach, web blog comment summarization stands out in the sense that multiple texts are being summarized. Each comment in the thread acts as its own text and each text can contain different viewpoints or ideas. This requires a slightly modified approach for extracting sentences in order to properly reduce redundancy while still gathering unique information about separate topics. A modified approach could also address the difficulty of multiple topics. A new approach for creating topically coherent and non-redundant summaries is presented in [11]. This work′s novel, topically coherent, approach identifies key concepts and relationships between these concepts. It builds a hierarchical relationship between the multi-document corpus, and chooses only the sentences with as little redundancy as possible. Other works such as [20], [21], and [16] create clusters of sentences and algorithmically produce individual topics by looking at sentences closest to the cluster centroid. These approaches are key in creating quality summaries in the context of web comment threads.

2.4 Query-Based Summarization

Next, we look at generic- versus query-focused summarization. Within text summarization, methods can either target their summaries using specific queries or topics or remain completely general. Several studies including [28, 34, 41] and more have focused on summarizing text based on certain topics using a query-based approach. These applications are mainly used in the scope of product review summaries, question-answering systems, and search, where the summary is generated with respect to the query. The query can either be input by the user, as in the case with search, be hard-coded, as a template to guide the summarization in different aspects as shown in [28], or be generated by using automated methods.

A preset number of hard-coded topics (queries) that split generic reviews into several categories has been exemplified in [28]. Instead of generating one summarized review for a product, there are multiple summarized reviews generated for different aspects of the product. The reviews are all based on the specific queries whether it be quality, battery life, or any other topic. While search takes a more supervised approach by requiring queries input by the user, both of these techniques have the downside of requiring a different set of queries for every summarization. Categories that might apply to one product do not necessarily transfer to others. While these approaches might work really well for product reviews and search, they do not necessarily transfer across different topics within comment threads on web posts.

Automatically generated topics can be very powerful in creating a summarization because it can generate its own set of queries based on certain topics within the comment threads. This makes the approach portable to threads with

different topics and creates a general solution.

2.5 Abstractive Summarization

After looking at a fairly large base of extractive techniques, we find that there is very limited research on abstractive approaches. This is mainly due to the fact that creating semantic representations of text and attempting natural language generation is very hard and still is a developing field within natural language processing. Current work in abstractive summarization has been delegated to mostly shallow methods and not true abstraction. These methods use a variety of techniques, some of the more popular ones rely on some form of sentence compression to generate novel text. One of the first approaches was based heavily on a templating approach. The system created was called SUMMarizing Online NewS articles, or SUMMONS ([31]); however, it required manually created templates, and picked the best words to fill in the given template-slots. According to [12], the SUMMONS system was problematic when used on larger domains where templates were harder to manually create. The work done in [27, 17, 15] is all based on some form of sentence compression techniques. While they do not greatly rely on true natural language generation, they have the advantage over extractive techniques in that they can form more concise sentences that do not have to show up in the original text.

One of the more interesting studies done in this segment is the Opinosis Graph-Based Summarization, [17], which is the basis of this work. Opinosis uses semi-shallow natural language processing, meaning it uses words strictly from the comment thread to generate novel sentences. This sidesteps the issue of natural

language generation while still creating novel sentences that do not necessarily appear anywhere within the text. Again, this approach is graph-based, except this time, it uses words as the indices, and the edges are now directional and connect to other words. Words that appear in the text are linked to the adjacent words creating sentence structures. The sentences in the text are tagged, and each word and its part of speech are used as unique nodes in the graph. So, when multiple sentences use overlapping words, those words get multiple linked edges allowing each unique word to only be represented by one node. Salience can then be measured by looking at subtrees and seeing the number of locations each node has been used. The main qualities that make the Opinosis Graph desirable are that it "naturally captures redundancies, it captures gapped subsequences, and it captures collapsible structures," as described in [17].

The Opinosis graph is adapted to handle a stream of varied, non-redundant opinions. Web comment threads contain multiple opinions, and are usually about several different topics. The key is to identify different topics together and to group the text. This would allow it to be handled by a modified version of the Opinosis Graph, which manages query-based sentences and uses a novel scoring approach. This approach will be expanded on and examined in further depth in the next chapter.

This is a promising technique that has yielded positive results; however, it does come with some drawbacks. This approach relies on a topically coherent graph, and it is used to summarize highly redundant opinions. This means the Opinosis approach is not particularly suited for large generic graphs for general summarization. When the graph gets dense and contains a large number of nodes, it yields a large number of sentences that are not necessarily coherent. Addition-

ally, while the Opinosis graph creates novel sentences not found in the original text, it relies on mixing the words from different sentences to create new sentences. This means summary sentences are not guaranteed to be grammatically correct or even necessarily coherent. The solution to all of these drawbacks will be explained in the following design and implementation chapters.

2.6 Validation Approaches

The nature of text summarization requires the summary to be judged by a human, since computers cannot yet correctly compute the quality of summarizations. This means that for a number of summarizations that my research produces, there will also have to be accompanying human-made summarizations. However, the amount of human work involved in the process can be minimized. The solution to this involves two metrics that have been created to analyze these types of problems: ROUGE and BLEU.

The BLEU, or Bilingual Evaluation Understudy, metric was introduced in 2002 with the purpose of automatically evaluating machine translation methods [33]. It aims to show that the closer a machine translation is to a professional human translation, the better it is. In order to get a "closeness" measure, the machine translation is compared to the human translation using a modified n-gram precision. Precision is calculated by summing all the n-grams from text A that show up in text B, and dividing it by the sum of all of the n-grams in text A. In order to penalize redundancy, the precision measure is modified to "clip" the number of redundant n-grams from A to the maximum amount of times those n-grams show up in text B. This metric works on unigrams, bigrams, or any

length n-gram, and has been shown to rate translations quite accurately.

In 2003, the BLEU metric used for machine translation was tested on assessing the quality of automatic summarization [25]. They found that the BLEU metric worked well; however, they chose to slightly adapt it. They decided to use a coverage score instead of a precision score, and use a brevity bonus instead of a brevity penalty because brevity in summaries is preferred as long as the content remains the same.

The model, ROUGE, or Recall Oriented Understudy for Gisting Evaluation, was then further enhanced in [24] and tested with several slightly modified versions of ROUGE: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. This work was then validated by comparing it to three years of manually labeled summarizations from the Document Understanding Conference (DUC). Virtually all of the methods outlined in this related works session use some form of ROUGE to validate their results. It has been proven to be a standard among research attempting to assess the quality of machine made text summarizations.

2.7 Tools Used

A couple of tools are used to help with the creation of the SPORK system. SPORK is written completely in Python. Python is well-suited and frequently used for natural language processing tasks. Additionally, several tools have been written in Python to aid common NLP tasks. One of the most popular tools is the excellent NLTK (Natural Language Toolkit) [9]. NLTK is frequently used throughout SPORK and provides a wide range of useful functions and built-in corpora that help simplify basic NLP tasks. An example of some useful func-

tions are sentence and word tokenization functions, machine learning algorithms, classifiers, clustering algorithms, and much more.

A common and important task in natural language processing is part-of-speech tagging. Part of speech tags are used extensively in SPORK and the pipeline relies on them to create accurate summaries. Although NLTK provides some built-in POS taggers, the Stanford POS tagger is widely considered one of the most accurate taggers [39]. The Stanford Log-linear POS Tagger is written completely in Java and uses the Penn Treebank tag set. The Stanford Tagger uses information from tags from both preceding and following words via a dependency network. Additionally, it uses several more fine-grained word features to help accurately tag words. Although the Stanford POS tagger is written in Java, NLTK provides a simple API wrapper that enables the quick generation of accurate POS tags. The problem of tagging and how it relates to SPORK is described in detail in Section 3.3.1.

Although several more algorithms and research are used in SPORK, the previously mentioned ones are obtained from online and directly used by the implementation. These tools are dependencies and must be installed before running the actual SPORK pipeline.

Chapter 3

Summarization of Reddit Comment Threads

Reddit is an interesting proposition for automatic text summarization simply because of the amount of data present. Again, as of this writing, Reddit is the 54^{th} highest ranked website by traffic according to [5]. The massive amount of traffic generates a vast amount of content in the form of posts and discussion comments. As one can imagine, the amount of content generated is simply too much for individual users to read through. Not only does Reddit provide a myriad of posts, but the comments associated with those post vary greatly. Multiple discussions occur within the same post, each containing different ideas and opinions. This poses a significant challenge to most current summarization techniques that do not explore these topics and only find the sentences that are ranked the highest. Some comments might respond directly to the original post while others diverge and are completely unrelated. The structure and nomenclature for Reddit are explained in detail in Section 3.1.

Using Reddit for automatic text summarization introduces some interesting challenges. The goal of this work is to create a framework that can overcome these challenges. A system, called **SPORK**, is devised to identify sentences that come from various topics and ideas within the unstructured text of Reddit. Summaries are generated from Reddit posts, which contain comment threads. Since a comment thread contains many comments, and each comment in the thread is considered a *document* (in terms of text summarization), a *multi-document* approach is taken. Additionally, the system relies on some prior knowledge of the data, creating a *semi-supervised* approach. Within the framework, queries are used to narrow-down and target the summary to specific topics, turning it into a *query-based* method. Lastly, the framework mixes in both *extractive* and *shallow-abstractive* techniques to ultimately create an extractive summarization. The focus on this work is not to create a well-formed, human-readable paragraph summary, but instead, to identify the ideas and opinions represented in the text. These ideas should be essential to the text and be crucial to its meaning. In summary, the proposed SPORK system uses a semi-supervised, multi-document, query-based, extractive approach to highlight sentences that support all of the key ideas present in the corpus. The rest of this chapter addresses this approach and explains the methods and reasonings behind each step.

3.1 Design Overview

The typical pipeline for an extractive text summarization approach begins by taking a corpus and tokenizing the text into sentences. This relates to both single and multi-document corpora. Once the sentences are tokenized, they can optionally be processed to optimize for different types of extraction analysis.

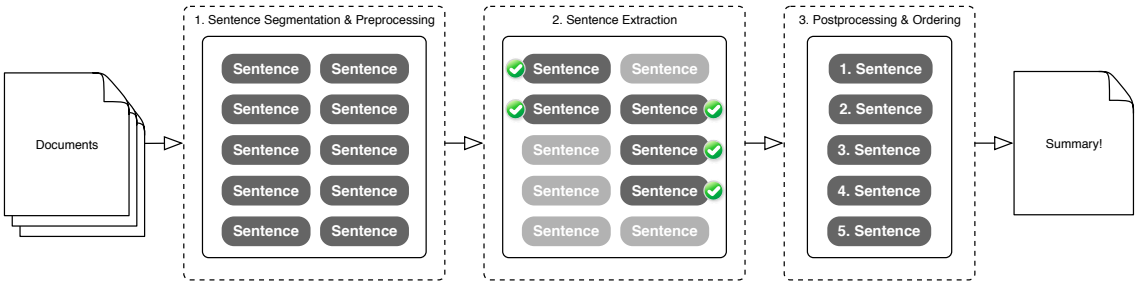

Figure 3.1: ***Basic Summarization Pipeline.*** **This figure shows the simplified pipeline of a basic extractive text summarization approach.**

A typical processing technique, sentence simplification, can include removing punctuation or stop-words. With clean, tokenized sentences, the next step is to extract salient sentences that can be used in the summary. We used a number of methods to extract relevant sentences (see Chapter 2). The last step of the typical summary process involves ordering and post-processing the extracted sentences. This includes making sure that topically coherent sentences are put together, and that punctuation is correct. The end result generally depends on intended format of the summary, whether it be for summarizing a product review, or actually making a coherent human-readable paragraph of text. Either way, in general, it is assumed to be a readable text that should adequately summarize the corpus.

As shown by the Figure 3.1, the steps are divided into three general blocks: preprocessing, extraction, and postprocessing. While this is a simplified overview of a general summarization approach, it illustrates the major steps required for most tasks.

Unlike traditional summarization methods, which usually just attempt to capture the most important sentences within the document, the approach designed in this work attempts to capture a summary based on different topics present in

a Reddit comment thread. This method tries to avoid the problems of redundancy present in other methods, and attempts to highlight the different ideas within the document. The work outlined in this research focuses on the first two major steps of Figure 3.1, *sentence segmentation & preprocessing* and *sentence extraction*, and aims to extract opinions located within different topics of the text. These steps are further broken up into sub-steps and are explained in more detail. Figure 3.2 illustrates the diagram of the new pipeline we call SPORK.

The SPORK system consists of three main steps: *data collection*, *preprocessing*, and *summarization*. The summarization step is based off of the Opinosis Graph [17] introduced in Section 2. Although this is an abstractive summarization approach, several issues arose with generating proper English sentences. The reasons for this are explained in later sections and are the main reason for switching to an extractive summarization approach. While the basis of the Opinosis Graph is still used in this work, we significantly adapt and extend it to be better suited for an extractive summarization approach. We then formally introduce SPORK as an extractive summarization pipeline well suited for the task of handling discussions with multiple opinions and topics such as the ones provided by Reddit.

The first thing to note about the SPORK pipeline in Figure 3.2 is that gathering the data is actually a vital step in the process. Additionally, the preprocessing and extraction steps have been expanded into several sub-steps which are discussed later in the chapter. By addressing a continuous stream of posts within a specific topic or category (ie. technology or politics), the abundance of information can be beneficial. The extraneous information from other posts in the category provides information on the relevance of keywords in a given post

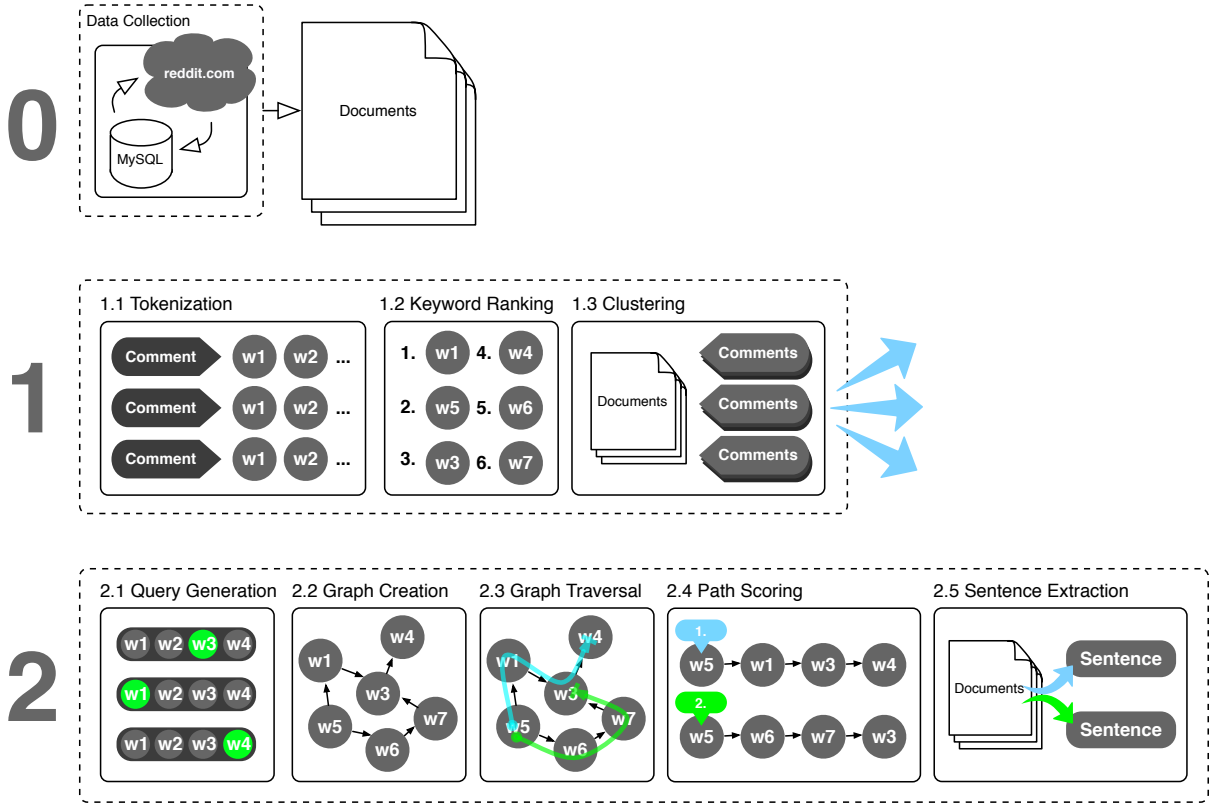

Figure 3.2: *SPORK Summarization Pipeline.* This figure shows the summarization pipeline suggested in this work. It focuses on the first two steps of the basic extraction method in Figure 3.1

using various methods such as TF-IDF, explained in more detail later. To begin, an explanation of the data source is required.

Reddit is an open source website for "what's new and popular on the web" [4]. Reddit is completely community-driven, meaning users upload and submit all the content. Links to articles are posted by users in specialized communities within Reddit, called subreddits. Articles can then be *upvoted* or *downvoted* by other users. This is used as an indication of the popularity of the article. Additionally, articles can be commented on by users, and those comments can then, in turn, be commented on, creating a tree-like structure of comments. Below is an outline of some of the basic terminology and information specifically related to the structure of Reddit articles that will be referred to throughout the rest of this paper.

- *Reddit articles* are usually referred to as *posts*. These posts usually only contain links to other websites or articles, and they only contain a title. A smaller subset of posts actually directly contains content; these posts are called self-posts. Figure 3.3 shows the structure of a Reddit post. Note the tree-like structure created by the nesting of comments in the comment-reply system.

- Only a small percentage of comments in any given post are direct responses to the article of the post. The rest of the comments are nested replies to other comments. The direct response comments are considered *top-level comments* and will be referred to as such in later sections.

- The *subcommunities* within Reddit where articles of similar topics are posted are called *subreddits*. Figure 3.4 shows a snapshot of the subreddit: */r/technology*. The subreddit is composed of posts, each containing the link

to the source and to its comments. The posts are sorted algorithmically using a combination of time since upload, number of pageviews, number of comments, and post score. Each subreddit has a list of rules determining which type of posts are allowed to be posted there. The right column of Figure 3.4 contains a list of posting rules as well as some statistics about the subreddit. Moderators are assigned to the more popular subreddits to ensure posting rules are followed. This subreddit regulation helps ensure posts are relevant and consistent with the topic at hand.

- *Upvotes* and *downvotes* sum up to create a *score* for each object. These will determine the order in which they are presented. This score is applied to Reddit posts as well as comments within that post. Shown in Figures 3.3 and 3.4, the arrangement of information presented is largely based on this score. Both figures show the scores of the comments and the posts respectively. In Figure 3.3, high scoring comments are shown first, and although the figure does not show it, comments with too low of a score are simply hidden.

This work uses material from a few of the most popular subreddits on Reddit: */r/Technology* and */r/Politics*. Each of these subreddits is ranked in the top 20 most active subreddits according to [6], with */r/Technology* accruing an average of 12,000 comments per day.

3.2 Data Collection

Since a semi-supervised approach is used in the SPORK pipeline, we collect several Reddit posts and their associated comments. Comments are collected

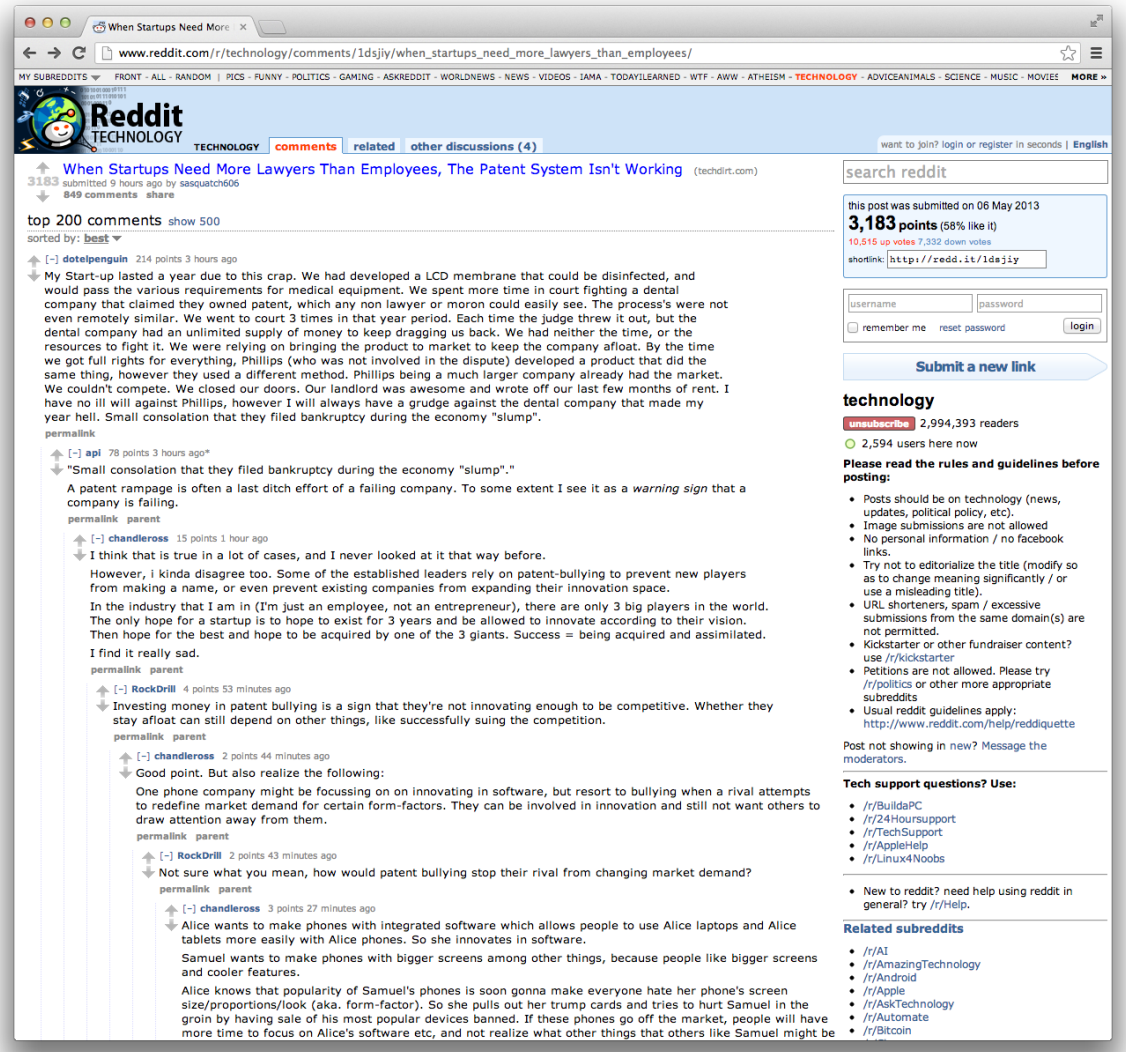

Figure 3.3: ***Sample Reddit Post.*** This figure shows the structure of a reddit *post*. This post comes from the */r/technology* subreddit and can be found at http://redd.it/1dsjiy.

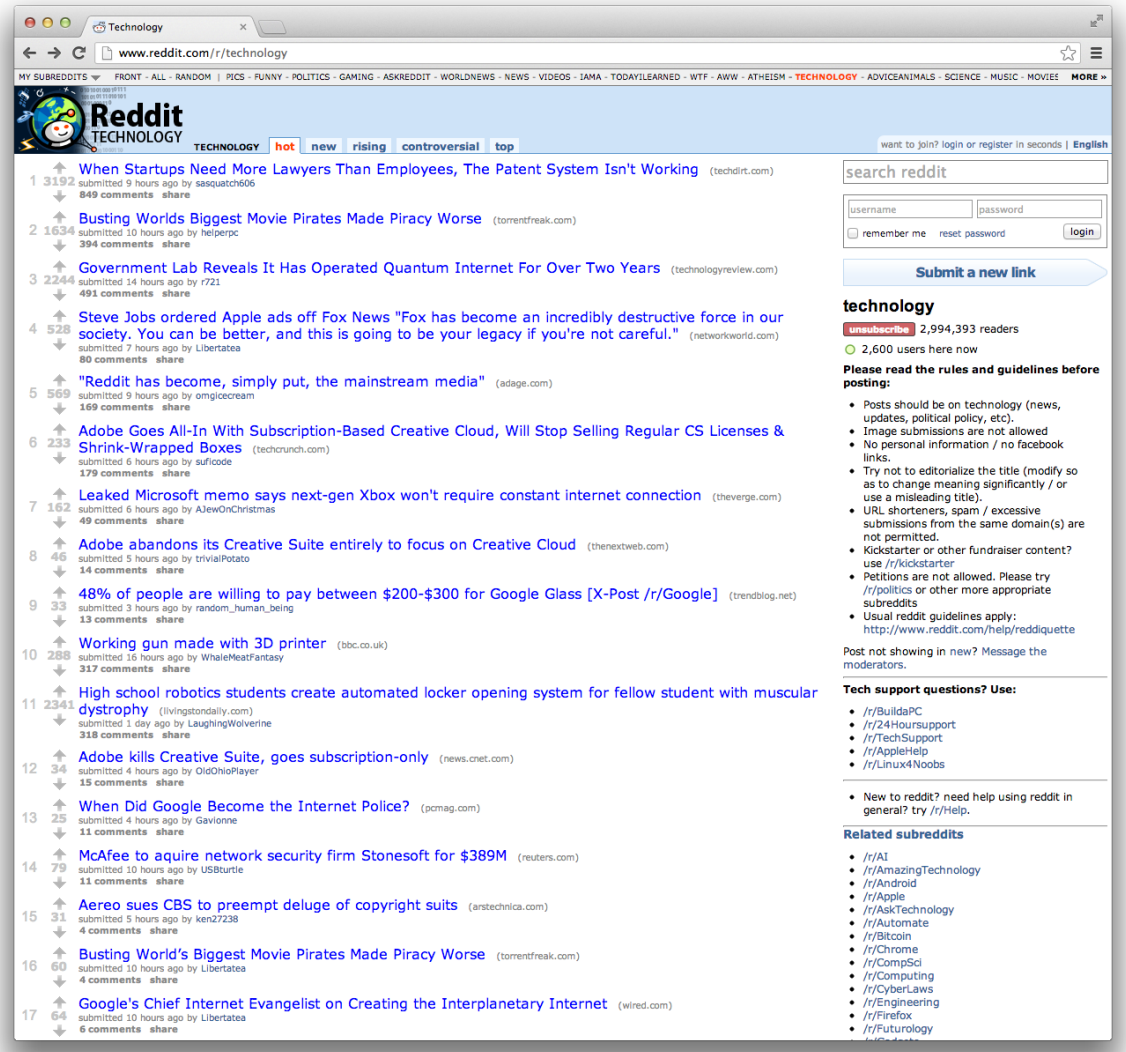

Figure 3.4: ***Sample Reddit Subreddit.*** This figure shows the structure of a typical subreddit. Specifically, the */r/technology* subreddit is displayed here.

from the two subreddits mentioned before, */r/technology* and */r/politics*. We do this to later evaluate SPORK on posts in these given subreddits. Since subreddits have similarities in common, it is useful to gather information from several posts within the subreddit. These similarities can be used to identify stop-words, as explained in Section 3.3.2. Our goal is to obtain enough comments within each subreddit to adequately identify stop-words. During our evaluation, we obtain tens of thousands of comments from each subreddit.

A running Python script begins the data collection process. We specify a subreddit, and the script begins collecting all new comments posted within that subreddit. These comments and all their attributes are saved and stored in a MySQL database. Reddit provides an API which can be used for querying data being changed on Reddit in real time [2]. Additionally, we use an open-source Python wrapper called PRAW, or Python Reddit API Wrapper, created in [3], to easily interface with the Reddit API in Python.

The API Reddit provides some restrictions in terms of accessing the information from their servers. They allow for a maximum of one request every two seconds or no more than 30 requests every minute [2]. This restriction prevents developers from simply querying their servers and pulling all past posts and comments from a subreddit. This is further complicated by the fact that it takes several requests to pull all comments within a single post. Since some posts contain thousands of comments, it takes much longer to query them compared to other posts with fewer comments. Therefore, instead of pulling all old data, only new posts and comments are acquired. Although this decision limits the amount of data pulled from Reddit servers, it continuously gathers data, potentially forever, while still obeying the Reddit API restrictions. Testing has shown that

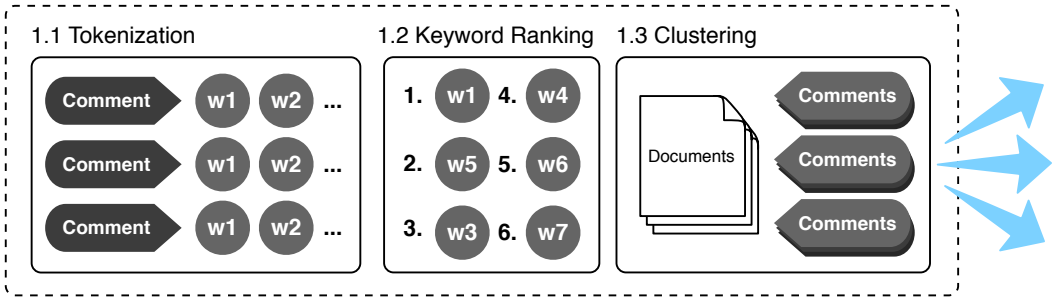

Figure 3.5: *Pipeline: Preprocessing.* This figure shows the preprocessing step of the summarization pipeline.

the limit of 30 requests per minute allows for pulling all new content within a subreddit, even on more popular subreddits such as */r/Technology* or */r/Politics*.

The rest of the steps in the pipeline use, as a running example, a sample Reddit post which is attached in Appendix B. The post helps illustrate the algorithms using a real example and is taken from the */r/technology* subreddit. A snapshot of the post was taken from: `http://redd.it/1cgncb` on May 10, 2013. At the time, the post contained 160 comments. Major portions of the web page are stripped to increase the readability of the post.

3.3 Preprocessing

The collected data is fed directly to the actual summarization process. The summarization process begins with some preprocessing required to help get a more consistent summarization. This part of the pipeline is illustrated in Figure 3.5, and contains a couple of steps.

The preprocessing proceeds as follows:

- First, comments are parsed and tokenized for use in the pipeline. This usually encompases tokenizing the comments into sentences and furthermore tokenizing the sentences into a list of words. Both of these tasks can be done using several different techniques, but the NLTK recommended tokenizers are used. Additionally, part-of-speech tags for the tokenized sentences are required later in the pipeline. The graph generation and traversing steps make heavy use of POS tags in Sections 3.4.2 and 3.4.3. Although NLTK provides several options for POS tagging as well, the Stanford Tagger, mentioned in Section 2.7, is used instead. The Stanford Tagger is a well tested and highly accurate parser [39]. In SPORK, the Stanford Tagger is trained using the left3words model: *wsj-0-18-left3words.tagger*. The *wsj-0-18-left3words.tagger* model is an order of magnitude faster then the standard WSJ22-24 test set and is nearly as accurate [18]m.

- Second, a keyword ranking approach is implemented to help find the importance of certain phrases and connections used later on in Sections 3.4.2, 3.4.3, and 3.4.4.

- Third, once keywords are ranked, the comments are clustered together into several different topics. This step helps streamline the graph creation and the traversing process by reducing complexity as well as maintaining more relevant graphs. These clusters of comments are the basis for the entire summarization generation seen in Section 3.4.

3.3.1 Tokenization and Tagging

The seemingly trivial task of separating sentences from a chunk of text, actually has its complexities. A simple sentence tokenizer would just separate sentences every time a terminating punctuation is encountered. The complexity occurs because periods do not have to terminate a sentence in the English language. Periods can be used for abbreviations, and, therefore, can be placed in the middle or at the end of a sentence. The NLTK recommends sentence tokenizer is the *PunkSentenceTokenizer*. It is an implementation of the multilingual sentence boundary detector created in [23]. It provides an unsupervised method that builds "a model for abbreviation words, collocations, and words that start sentences; and then uses that model to find sentence boundaries [1]."

Similarly, segmenting words within sentences comes with its own set of difficulties. The difficulties stem from finding the appropriate word boundaries, such that the words are properly segmented. This becomes difficult when taking into account contractions, commas, and other punctuation. The recommended NLTK technique uses the *TreebankWordTokenizer*. This tokenizer follows several steps regarding punctuation and uses regular expressions to tokenize text as in the Penn Treebank [29].

Finally, we use the Stanford POS Tagger to accurately tag the tokenized text. POS Tagging is essential to numerous natural language processing tasks. Difficulties in POS tagging can be attributed to the ambiguities of the English language. Not only do words have different part-of-speech tags in different contexts, but sometimes words can have multiple meanings. Thanks to the ambiguities of the language, sentences can therefore have multiple meanings and be interpreted differently. Using a classic example, take the sentence: "*I made her duck*". This

sentence can be interpreted in several different ways. For example, it could mean that I actually cooked her the animal duck. Alternatively, it could mean that I forced her to lower her head. There are several more ways this sentence can be interpreted, but the point is that ambiguity exists continuously, which can even make it difficult for humans to tag words. Because of this, generating a set of rules to determine parts-of-speech for words can be very difficult. To attempt to solve this problem, information from the surrounding context of the word can be used to help. These lexical features can include looking at the words directly behind and in front of the given word as well as several others. Probabilistic models for POS taggers can also increase the accuracy by providing the most likely tag for a given word based off of information provided by a corpus. The Stanford tagger uses all of these techniques and several more advanced ones to achieve accuracies of 97.24% using the Penn Treebank [39].

3.3.2 Keyword Ranking

The next step in the pipeline is ranking keywords (see Figure 3.2). This step requires tokenized sentences and words and aims to rank the importance of all keywords in the comment thread. This is done by obtaining the Term Frequency-Inverse Document Frequency (TF-IDF) rank for each word. TF-IDF defines the relationship between the number of times a word shows up within a specific document to the inverse of the number of other documents the word shows up in the corpus. The idea is if the word shows up in a document several times, it must be important. However, a word that shows up multiple times could simply be a common word, a stop-word, such as “the” or “a”. To offset the high frequency of common words, we balance the score by weighting it with the inverse

document frequency. This suggests that words that show up in several documents are common to all documents and, therefore, are not significant. Intuitively, this makes sense. If a word shows up a significant number of times in a document and, at the same time, does not show up in many other documents, it can be considered a significant keyword for that given document. An overview of the general equation for getting the TF-IDF score of a word can be seen in Equation 3.1. In this equation, the word w is found in post, d, which is in a set of posts, D, such that $d \in D$ [38].

$$tfidf(w, d, D) = f_{w,d} * \log\left(\frac{|D|}{f_{w,D}}\right) \tag{3.1}$$

The $f_{w,d}$ represents the frequency of the word in the chosen document. $|D|$ represents the total number of documents in the corpus and $f_{w,D}$ represents the number of documents in which the word shows up. This score is simply the product of both individual measures: 3.2 and 3.3.

$$tf = f_{w,d} \tag{3.2}$$

$$idf = \log\left(\frac{|D|}{f_{w,D}}\right) \tag{3.3}$$

Different keyword ranking methods are used for single and for multi-document corpora. However, since comments are collected from several posts in a subreddit, a method that takes advantage of the extra information can provide better results. As explained above, the IDF portion of the score ensures that common words are not given high ranks. This does not necessarily mean that only simple common words, such as "the", are affected, but more complex words that are common to the corpus. This is the benefit of using TF-IDF which is used for normally used in

NLP for filtering content-words. An example illustrating this can be easily seen when looking at different subreddits. Within the */r/technology* subreddit, words common across multiple documents might be "technology" or "computer", while common words in the */r/politics* subreddit could be "republican" or "democrat". These words will get lower ranking scores in their respective subreddit making it much more accurate than a simple stop-word list.

TF-IDF is by no means the only way to rank keywords or to find stop-words, but it has been proven to be simple and effective [37]. The concept behind TF-IDF weighting is fairly simple and has been used for a long time [38]. There have been many variations to finding TF-IDF scores as seen in [37]. Some enhancements have been added to the basic TF-IDF method to increase performance or increase accuracy. Lemmatization or stemming ensures that words such as "house" and "houses" will be grouped together for frequency counts. When we use this type of approach, we ensure that derivations of the same word count frequencies together.

The score for every unique keyword in the comment thread is stored and used for ranking summary paths later, in Section 3.4.4.

3.3.3 Clustering

When using the graph-based summarization approach it is beneficial to have redundant overlapping opinions. This is tough since the Reddit comment threads contain so many different discussions and opinions. In order to improve the efficiency of the summarization approach we want to group certain comments in a given comment thread together. This focuses the discussion into more manageable and cohesive groups. We achieve this by clustering like-comments together. This step is highlighted in Figure 3.3 and the benefits and reasoning behind

clustering are explained in more detail in Section 3.4.2.

The tree structure of the comment thread can help provide some insight into how comments should be clustered. *Top-level comments* are seen as direct responses to the post article, while all non top-level comments are simply responses to top-level comments. This allows us to only worry about how the top-level comments should be clustered together. All non top-level comments are simply responses to the topics and join the known clusters once they are defined. On average, the number of top-level comments in a post ranges from 25 to 80 on a 200 comment post. This means the number of clusters as well as the size of the clusters vary from post to post.

To address this problem, we use a hierarchical clustering approach to combine similar comments. Hierarchical clustering can either be done *agglomeratively* or *divisively*. Agglomerative clustering is fairly straightforward to implement and uses a bottom-up approach [26]. The pseudo code can be seen in Algorithm 1. Initially, each comment is treated as its own cluster. A similarity matrix is then computed for each of the clusters based off of a given similarity measure. The two clusters with the highest similarity are then merged together if their similarity passes an empirically set threshold. This loops until there is either one cluster left, or the clusters are not similar enough to merge. The clusters are stored in a Dendrogram data structure. This structure is really a labeled binary tree with a few additional properties that make it well-suited for the task. It manages the merging of clusters and, depending on the implementation, can also automatically split the dataset into clusters based off of a similarity threshold.

As opposed to some non-hierarchical clustering techniques (k-means), comments are merged based off of an empirically determined similarity threshold

instead of a predetermined number of clusters. This is important because each post contains a different number of comments and, therefore, not all posts will contain the same number of topics or clusters. To find the ideal similarity threshold, several different posts were tested with varying numbers of comments.

Algorithm 1 Agglomerative Hierarchical Clustering

```
procedure CLUSTERING(c ∈ P)                    ▷ For every comment c in post P
    for all c ∈ P do                           ▷ Assign each comment to its own cluster
        clusters.add(cluster(c))
    end for
    while clusters.length > 1 do
                                               ▷ Compute the similarity matrix between all clusters
        matrix ← computeSimilarityMatrix(clusters)
        score ← maxSimilarity(matrix)          ▷ Get the most similar clusters
                                               ▷ Merge, only if above threshold
        if score > SIMILARITY_THRESHOLD then
            clusterPair ← maxSimilarityClusters(matrix)
            clusters.merge(clusterPair)
        else
            return clusters                    ▷ Return when similarity does not pass threshold
        end if
    end while
    return clusters                            ▷ Or return when there is only one cluster left
end procedure
```

There are several different ways to compute the similarity between comments. Since comments are essentially a collection of sentences, sentence similarity approaches can be applied to the task. Sentence similarity measures are used in

several natural language processing applications. Several additional sentence similarity measures are explored later in Section 3.4.5, any of which can be used. Although there are some more advanced measures for the task of clustering comments, for the sake of brevity, a cosine similarity measure is used. Cosine similarity provides a similarity measure between two vectors. In the case of comment or sentence similarity, the vectors are a list of words. We generate comment vectors out of the each comments using the tokenization techniques from Section 3.3.1. Given two vectors A and B each representing the words in comments a and b respectively, the general formula for cosine similarity can be seen in Equation 3.4. The dot product of the two vectors is divided by the product of the magnitudes of each of the vectors.

$$cosine\_similarity = \frac{A \cdot B}{|A| * |B|} \tag{3.4}$$

This cosine similarity measure is used to compute the similarity matrix in the hierarchical clustering algorithm. However, the algorithm requires clusters being compared to other clusters, each potentially containing several comments, while the cosine similarity measure only takes into account two comments. In order to properly measure this, we use one of the well-known methods below.

- **Single-Link Method** - This method measures the similarity between two clusters based on the two single most similar elements within the cluster.
- **Complete-Link Method** - This method measures cluster similarity based off of the two most dissimilar elements within the cluster.
- **Average-Link Method** - This method averages all similarity scores of every element in the first cluster to every element in the second cluster.

We use the single-link method in this work as it resulted in the best clusters. Using the single-link method along with the cosine similarity measure, the $SIMILARITY\_THRESHOLD$ parameter was determined to be .15. This clustering technique is important to the graph creation and traversal stages of the pipeline. Each of the subsequent chapters in this section are repeated for all of the clusters generated with this technique. This means that we execute step 2.1 through 2.5 of Figure 3.2 on all of the clusters generated in step 1.3.

Once all top-level comments are clustered, the remaining non top-level comments are assigned to one of the existing clusters. This is done by finding the largest similarity between each remaining non top-level comment and the calculated set of clusters. The similarity is measured using the same single-link cosine similarity approach as in the hierarchical clustering algorithm.

Finally, clusters that contain fewer than of 15 comments are seen as insignificant and thrown out. Although these clusters might contain unique opinions, we only want to gather the really important ones. If each comment in a comment thread represents a unique opinion and we try summarizing with all clusters, the summary generated would not be any shorter than the original text. Therefore, we assume that clusters that do not meet a minimum redundancy of opinions should be thrown out to control the length and quality of the summary. This threshold has been empirically set to 15 as the graph-based summarization approach runs best when there are enough redundancies in a text.

Example. For our running example, mentioned in Section 3.2, clustering all 160 comments from Appendix B, resulted in 5 unique clusters. Out of the 160 comments, this thread contains a total of 20 top-level comments. These 20 top-level

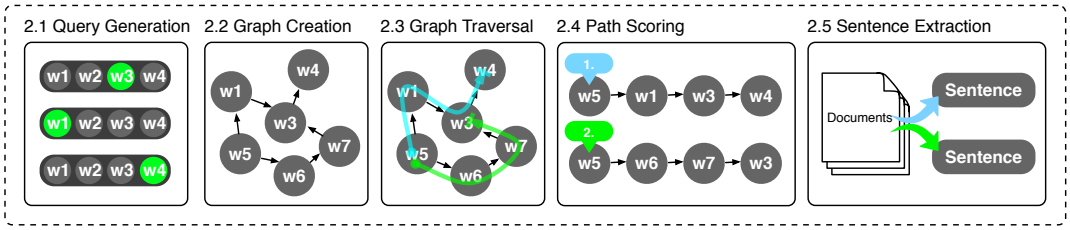

Figure 3.6: ***Pipeline: Summary Generation.*** **This figure shows the summarization step of the pipeline.**

comments are condensed into 9 clusters while the remaining 140 are merged into them. Clusters that contain fewer than 15 comments are discarded. Ultimately, there are 5 remaining clusters, broken up as such:

Cluster	Cluster Size (In Comments)
1	30
3	18
4	30
5	42
7	16

Table 3.1: ***Comment Clustering Example.*** **This table shows how the clustering algorithm runs on the sample Reddit comment thread provided in Appendix B.**

3.4 Summary Generation

This section of the pipeline is broken up into several steps, represented in Figure 3.6, and deals with the actual summarization generation. It is based off of a modified version of the Opinosis Graph, introduced in [17]. As explained in Chapter 2, there are several properties of the Opinosis Graph that make it a good candidate to use for summarization. There are, however, several issues and limitations with it that are addressed in the following graph generation and

graph traversal sections, 3.4.2 and 3.4.3, which illustrate the need to adapt the algorithm.

The process of this section of the pipeline is as follows. First, we generate queries to represent important ideas contained in the comment clusters. Then we create a modified version of the Opinosis Graph, and traverse it targeting the query keywords. The traversed paths are then scored and ranked based on a novel algorithm. Finally, we use the highest scoring paths to extract similar sentences from the comment clusters. We find the similarity scores of the sentences in the comment cluster to each of the high-ranking paths. The path-sentence combinations with the highest similarity scores are chosen for the summary. This process repeats for all of the comment clusters generated in Section 3.3.3.

3.4.1 Query Generation

After the comments have been clustered into topics, representative keywords are extracted for each cluster to target the graph traversal. The technique used to create and traverse the graph using these queries is explained in the next sections. Basically, while the Opinosis Graph is powerful, it only supports highly redundant phrases or sentences. To circumnavigate this issue, queries highlighting the important topics of the cluster are generated to better target the graph.

The approach used to generate queries is based off of the χ^2-measure, which is frequently used in statistics. The χ^2-measure is often used to compare some observed data with the expected outcome of that data. The observed, expected data is measured in the form of some frequency of a given event. The basic statistical value of χ^2 is defined as

$$\chi^2 = \sum \frac{(F_o - F_e)^2}{F_e}. \tag{3.5}$$

Here, F_o represents an observed frequency of the event while F_e represents an expected frequency of the event. This value measures the *bias* between a set of expected frequencies and observed frequencies. The work done in [30] outlines a well-tested approach for using the bias of co-occurrences of words in sentences to extract meaningful keywords from a text. The process is broken down as such.

1. All terms in a given corpus, $w \in W$, are extracted and the most frequent ones are stored, $g \in G$. The number of frequent terms used can vary, but our case, the top 30% of all terms are used.

2. A co-occurrence matrix is built, identifying the number of co-occurrences of every term with the subset of all frequent terms defined in the first step. Co-occurrences can be identified in a number of ways, in our study, a co-occurrence between two terms exists if they show up in the *same sentence* together.

3. If any term statistically tends to co-occur more with only a subset of the frequent terms it is said to be *biased* towards that subset of frequent terms. This bias is measured with the χ^2 measure and determines the importance of the term. The higher the bias, the more likely the term is regarded as important.

In terms of measuring the χ^2 value with respect to a word co-occurrence model, the equation is defined as

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g}. \tag{3.6}$$

To calculate the χ^2 value of a term w, the same general χ^2-measure is used as in Equation 3.5. In the case of Equation 3.6, $freq(w,g)$ is the frequency of co-occurrence between term w and frequent term g; this represents the observed frequency. The expected frequency of the measure is then calculated as the product of n_w and p_g. p_g can be broken down into the total number of terms in sentences where g appears divided by the total number of terms in the document, shown in Equation 3.7. Finally, n_w is the total number of terms in documents where w appears.

$$p_g = \frac{\sum \#\text{ of terms in sentences where } g \text{ appears}}{\#\text{of terms in document}} \tag{3.7}$$

This measure is then modified to increase "the robustness of the χ^2 value" [30]. Important terms are identified when a term co-occurs multiple times with a frequent term. Sometimes, this incorrectly identifies important words because non-important words can be paired with important words. The terms "strong" and "weak" tend to co-occur with the frequent term "signal" because they show up in the form of "strong signal" and "weak signal". The two terms show a bias towards a frequent term and therefore get high χ^2 values even though they are not important keywords. Using the modified equation shown in Equation 3.8, the χ^2 value will be lowered if a term tends to only co-occur with **one** frequent term. The maximal term is subtracted from the overall score greatly reducing the χ^2 value of the two terms. However, terms that co-occur with several frequent terms will still maintain high χ^2 scores.

$$\chi'^2 = \chi^2 - \max_{g \in G} \left\{ \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \right\} \tag{3.8}$$

Furthermore, the measure is enhanced by adding some additional preprocessing steps. The first step simply removes stop-words based off of the SMART System stop-list provided in [22]. The work in [30] is then further extended to include additional preprocessing steps, such as the Porter Stemming Algorithm [35], a phrase detection system and a keyword clustering algorithm. For the sake of simplicity, these additional features are not included in our study nor are they necessary for the types of keywords we generate.

Keywords that have high χ^2 scores are not necessarily the most frequent terms in the cluster. This is as expected since the measure is not a frequency counter and is made evident in the following example. However, since these queries are fed to the graph to narrow and target the traversal, the queries must occur enough times in the cluster for them to be traversable. To solve this, the top 10 terms ranked by χ^2 are generated for each cluster, and the *two* with the highest frequency count are used as query words. This ensures that not only the keywords have high χ^2 values, but they also occur frequently enough to produce meaningful traversal paths. The more χ^2 queries used, the more output sentences are generated for the summary. Using two query words results in two sentences being generated per comment cluster. For the running example, two query words are used for each cluster.

Example. Running the χ^2-measure on the previously generated clusters of the reference post in Appendix B, some important keywords were extracted. Out of the seven clusters automatically generated in Section 3.4.1, queries for the largest

3 clusters are tabulated. The top 5 important keywords were found and shown in Table 3.2.

Cluster	Frequency (In Cluster)	χ^2 Rank	χ^2 Value	Term
1	10	5	3,158	**performance**
1	6	7	3,030	**power**
1	6	9	2,557	parallel
1	5	8	2,574	true
1	4	1	16,515	deleted
4	6	1	7,393	**thing**
4	5	4	6,086	**core**
4	5	5	5,907	power
4	4	2	6,173	cores
4	3	3	6,089	regular
5	29	1	47,583	**cores**
5	16	3	28,693	**people**
5	14	7	20,773	parallela
5	12	6	21,664	point
5	9	2	38,810	future

Table 3.2: *Query Generation Example.* This table shows how the query generation algorithm running on the 2 largest clusters of Table 3.1. The sample Reddit comment thread, provided in Appendix B, is used to extract the clusters and queries.

Although only the top two query words are used actually used, Table 3.2 shows the top five to better demonstrate the algorithm. The first thing to note is that the terms with highest χ^2 ranks are not always the most frequent keywords. As explained before, out of the top 10 ranked terms, the ones with the highest frequencies are chosen first. This ensures the that the graph is properly traversable. Generally, the clusters with more comments generate higher χ^2 scores and higher frequencies, which, in turn, leads to more graph traversals (as shown in Section 3.4.3). Finally, some overlap of keyword terms occurs within the different clusters. This can either mean that there is some overlap between two clusters, or more likely, that the clusters use the keyword terms in different

contexts. The overlap is minimal and should never produce the same sentence since the clusters are isolated.

3.4.2 Graph Generation

We use the tokenized and tagged clusters of comments to create a modified version of the Opinosis Graph presented in [17]. As explained before, the Opinosis Graph uses a graph structure to represent natural language text with nodes and edges. We then generate summaries by traversing the nodes following the edges. This creates paths from the graph that represent sentences. Since these sentences are not necessarily found in the original text, this approach is seen as a sort of shallow-abstractive summarization. There are several important properties to the graph that help define its structure.

Nodes within the graph represent the words in the cluster of comments while edges between the nodes represent the connections between words in the sentences of the comments. This is similar to the concept of a word lattice, except the graph is not guaranteed to be acyclic, and the words represent the nodes. Furthermore, each word accompanied with its part of speech is used as a unique identifier string for a node in the graph. This uses the tokenization and part-of-speech methods from Section 3.3.1. Remember, the POS tagger used is based off of the Penn Treebank which contains 36 unique part-of-speech tags. Consider, for example, a sentence found in one of the comments of the post in Appendix B:

"Knowledge is not illegal."

After tokenization and using the Stanford Tagger, the sentence becomes:

"Knowledge/NNP is/VBZ not/RB illegal/JJ ./."

Five unique nodes are created for this sentence. The unique identifier combines the word with the part-of-speech delimited by a colon. For example, the unique identifier string for the node created of the first word in the sentence is *"knowledge:NNP"* (all words are normalized to eliminate duplicate nodes that contain the same word with different letter cases). Since there are many words in a cluster of comments, there are bound to be repeats of the same unique identifiers. The duplicate information is important and is used to capture redundancies in the graph later on. In order to capture this information, each node contains a list of meta-information objects. A meta-information object represents one instance of a word in the comment cluster. Each meta-information object stores several things about the instance of the word. The fields within the meta-information object are detailed in Table 3.3.

Field	Description
word_id	Position of the word within the sentence it was found.
sentence_id	Sentence number within the comment in the document.
comment_id	The id of the comment within the cluster.
author_id	The id of the comment the word was found in.
score	The score received by the comment containing the word.

Table 3.3: *Meta-Information Object Fields.* The fields of the meta-information objects stored in each node of the graph.

Edges between the nodes then represent the connections between words in the sentences of the document. In this case, the node with the unique id *"knowledge:NNP"* would point to *"is:VBZ"*. Redundancies in the nodes simply increase the number of outgoing and incoming edges to each node. Theoretically, the node *"is:VBZ"* would contain several outgoing links to other nodes since it is a fairly

common word and part-of-speech combination.

The algorithm for creating the graph is based on the one presented in the Opinosis Graph [17]; however, it is modified to include several additional pieces of information provided by the comment thread. The pseudocode for the modified version of the algorithm is presented in Algorithm 2. In their algorithm, positional information, analogous to the meta information here, is much simpler and stores only the *word_id* and the *sentence_id*. Additionally, clusters of comments are parsed instead of simply parsing a set of sentences. As shown in Algorithm 2, a triple nested loop iterates through comments in clusters, sentences in comments, and finally, words in the sentences. Graph nodes are then created for every word. If a given word already has an associated node, the meta information of that word is simply added and the node is not recreated. Finally, edges between nodes are created only once between two unique nodes.

We demonstrate the algorithm and graph creation on a very simplified example. A select few comments from Appendix B, are illustrated in Table 3.4. These comments have been simplified to better illustrate the graph.

Comment	Text
Comment 1	It's basically a Raspberry Pi except it's a lot more powerful.
Comment 2	Is this just a Raspberry Pi?
Comment 3	Is this better then the Raspberry Pi?

Table 3.4: *Simplified Comments For Sample Graph Creation.* Three comments taken from the sample Reddit post in Appendix B which are used to create a sample graph in Figure 3.7. The comments are simplified to better visually present the graph.

As shown in the graph generated in Figure 3.7, the edges are directional and are created by using the structure of the sentences. Each node with multiple

Algorithm 2 Opinosis Graph Creation Algorithm

Input: Tokenized and tagged comments in a cluster: $C = \{c_i\}_{i=1}^{n}$

Output: $G = (V, E)$

procedure GRAPH(C) ▷ For every comment c in cluster C

for $i = 1 \rightarrow n$ **do**

$num\_sents \leftarrow SizeOf(c_i)$

for $j = 1 \rightarrow num\_sents$ **do**

$sent = c_{ij}$; $sent\_size \leftarrow SizeOf(sent)$

for $k = 1 \rightarrow sent\_size$ **do**

$unique\_id \leftarrow GetUniqueId(sent_k)$

$author \leftarrow Author(c_i)$

$score \leftarrow Score(c_i)$

$meta\_info \leftarrow MetaInformation(k, j, i, author, score)$

if $ExistsNode(G, unique\_id$ **then**

$v_k \leftarrow GetExistingNode(G, unique\_id)$

$AddMetInformation(v_k, meta\_info)$

else

$v_k \leftarrow CreateNewNode(G, unique\_id)$

$AddMetInformation(v_k, meta\_info)$

end if

if $notExistsEdge(v_{k-1} \rightarrow v_k, G)$ **then**

$AddEdge(v_{k-1} \rightarrow v_k)$

end if

end for

end for

end for

end procedure

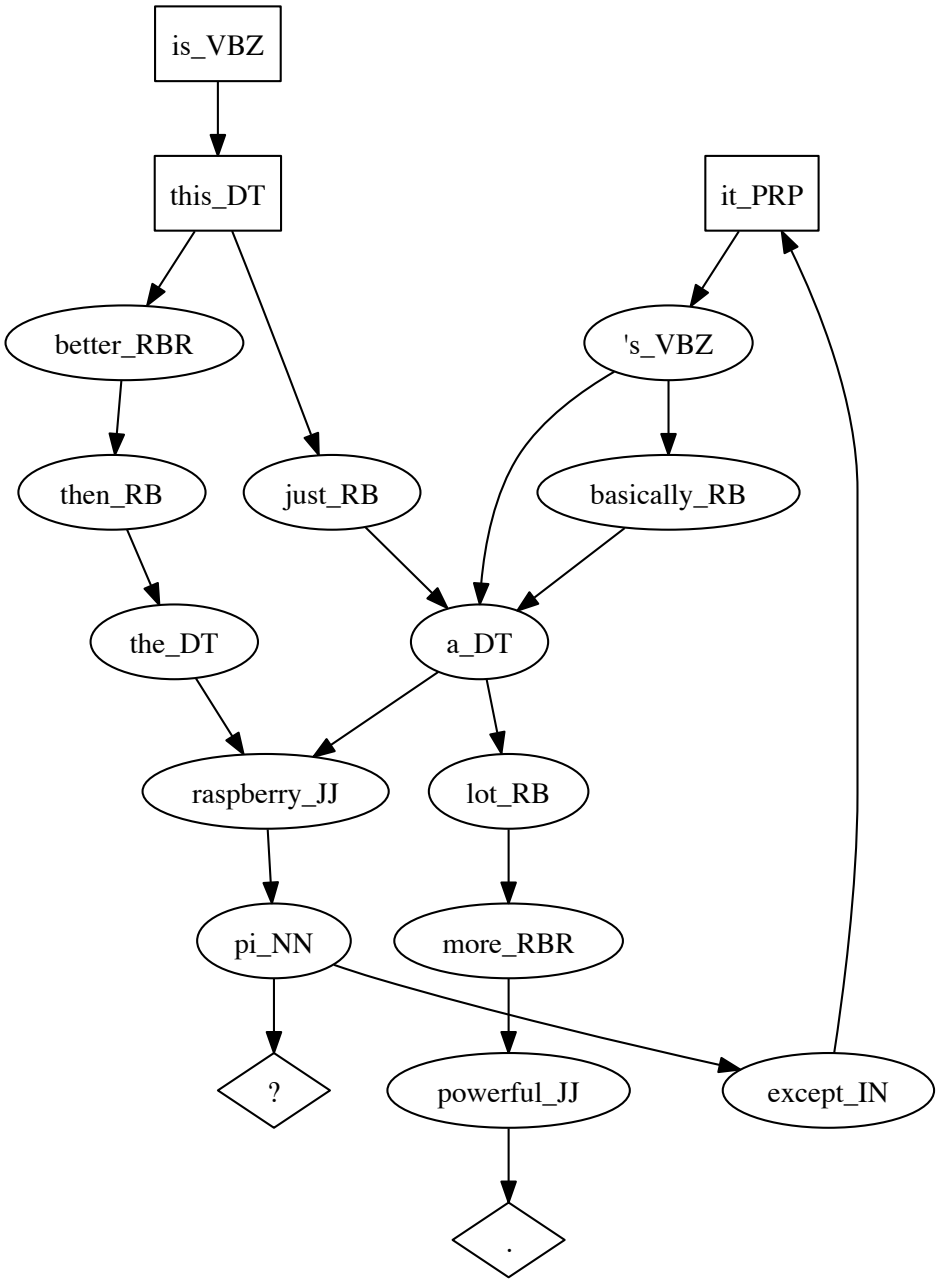

Figure 3.7: ***Sample Graph Creation.*** **This figure shows a sample graph created from the comments in Table 3.4.**

links contains all of the meta-information objects from each instance of the word. This graph has a couple properties that make it a good tool for summarization. First, the graph naturally captures redundancies from the original document. Since the nodes contain a list of all of the meta-information objects, we can use that to gage the strength of the connection between nodes. Not only that, but we can find gapped subsequence between connections. This means that even if two words are not directly next to each other, a connection can still be found. An example of this can be seen from the following two sentences: "The screen was bright." and "The screen was very bright". Even though the words "was" and "bright" do not appear directly next to each other in the second sentence, the connection is still found. In the Opinosis Graph, this is determined by a *MINIMUM_GAP* parameter that specifies the maximum number of words that can be found between two words for those two words to still have a meaningful connection. In addition to these properties, the graph can also compress certain text by finding so called "collapsible structures". Although, as this is not the main focus of this work, it can be seen as an extension of future possibility to improve the quality of summarizations.

Although the Opinosis Graph provides some great properties, the creators of the Opinosis Graph used very redundant opinions as the input to the graph. This can lend itself well to the use of product reviews that are all about the same thing. In their experiment, reviews from hotels, cars, and various other products were used. Manual work was required to ensure that all of the reviews from a specific product would all correspond to a singular topic or opinion. In that case it meant constructing queries that tied in the name of the product and the specific aspect of the product the user was interested in. For example, if

Amazon contains several reviews about a specific product, a query was manually constructed to include the name of the product (ie. Amazon Kindle) with the aspect of interest (battery life). The query is then used to simply filter out all reviews that do not contain all of the words in the query. This ensures that all of the remaining reviews at least contain the words that have been manually determined to be about a specific topic. There are several downsides to this solution. First, the need to manually create queries reduces the portability of the approach and does not apply itself to a general summarization case. Additionally, the filtering approach does not encapsulate all reviews about specific topics. In many cases, different words can be used to describe the same topic. For example, *screen resolution* and *PPI* (another term relating to pixel density) can describe the same idea. Unfortunately, the technique used in the Opinosis Graph has no way of understanding this.

When simply trying to apply the approach outlined in the Opinosis Graph, several issues arose. The nature of a comment thread leads to multiple opinions and even different subtopics about the article. This means the data is much less structured, which in turn, makes it harder to generalize. Because of this, the resulting graphs created from this data looks much different then the graph created by using redundant product reviews, as in the Opinosis Graph. With highly redundant opinions, fewer unique nodes are created due to the fact that a lot of the reviews contain similar words with a similar structure. Not only that but, the connections, or edges, between the nodes are much more structured reducing the overall complexity of the graph.

This is important because using the Opinosis Graph on an unstructured comment thread with around 150 comments leads to immense complexity when

traversing. An example comment thread with 200 comments with an average of 40 words per comment contained around 5,000 unique nodes in the graph. Traversing through all of these nodes each contained a multitude of connections lead to around four hours of runtime. However, more problems arose when simply trying to reduce the number of comments. When producing results from an unstructured comment thread with fewer comments, the resulting paths were completely incoherent. This is due to the fact that the node connections were coming from sentences or comments of different topics. When the paths were traversed, many connections came from different topics leading to gibberish sentences.

In order to address these issues, we modify the graph approach and add the additional steps outlined in Figure 3.2. To make a more cohesive graph, we use the clusters created in Section 3.3.3 as the input data for each graph. These clusters are then combined with the χ^2 queries we generated in Section 3.4.1. Both of these help ensure that the sentences are more focused on a more narrow topic and ensure that noise is reduced from other opinions. We explain this to help justify the design choices made about clustering and query generation and we further explore it in Section 3.4.3.

Example. Continuing the running example from the previous two sections, graphs are generated for the top three clusters. Due to the complexity of the graphs, they cannot be represented visually. Instead, the parameters and output of the graphs are described in Table 3.5. While the first two clusters contain approximately the same number of words and nodes, the last cluster contains significantly more. It is also noted that due to this, the query frequencies are much higher as well as the number of generated sentences shown in Section 3.4.3.

Cluster	Comments	Sentences	Words	Unique Nodes
1	30	87	1595	569
4	30	95	1572	624
5	42	247	4867	1218

Table 3.5: *Graph Generation Example Output.* This table shows the output of the generated graphs of the 3 largest comment clusters created in Section 3.3.3.

3.4.3 Graph Traversal

The next step in the process is traversing the graph and finding paths that contain useful information. As discussed in the previous section, there are many properties of this graph that enable it to capture the important parts of the text. In order to actually find sentences, the graph must be traversed.

Traversing the graph enables us to find paths that chain together words to possibly form sentences. To achieve this, a couple steps must be followed in order to produce something meaningful. As explained in the Opinosis work, the traversal must contain the following.

1. **Valid Start Node (VSN)** - A valid start node is any node in the graph that has in some instance been marked as the first word of a sentence. In other words, at least one of the meta-information objects attached to the node must have a word-id of 0. These nodes are represented in Figure 3.7 by a square box.

2. **Valid End Node (VEN)** - Similar to the VSN, the valid end node is any node that has been used to terminate a sentence. VEN's are represented with a diamond in Figure 3.7.

3. **Valid Path** - The valid path rule ensures that the path follows some sort of structured sentence outline. This rule varies the most and requires some manual testing to get right. The Opinosis authors suggest to use some sort of regular expression POS matching, but this constricts the outcomes of the sentences.

We refer to the Opinosis Graph presented in [17] as the "basic Opinosis Graph" and all the algorithms it describes we refer to as the "basic Opinosis Graph algorithms". The basic Opinosis Graph traversal algorithm starts by iterating through all valid start nodes. For each valid start node, the graph is traversed, and the outgoing edges are followed. Every path is explored that leads from a valid start node to a valid end node. This is a recursive function that keeps track of the paths along the way. Once a valid end node is reached, the path is checked for validity. If the path is valid, it is stored for later scoring. This basic algorithm is altered with the addition of the query keywords generated in Section 3.4.1.

Instead of starting at all VSNs and traversing to VEN's, for each query, the paths *must* contain the keywords in the query. This means that a lot of the paths will no longer be relevant and should be thrown out. To make the algorithm more efficient, we start the graph traversal at the the query keywords. The traversal then goes in both directions. We begin by creating a list of backwards propagations, by starting at the keyword, we traverse in the reverse order and generate all possible paths until VSN is hit. We then generate a list of all forward propagations, starting at the query keyword and traversing forwards until we reach a VEN. Each of these propagation directions still follow the rules of the basic Opinosis Graph and only follow graph edges that contain the proper

number of redundancies. The redundancy between two nodes is defined as the number of times the words from the two nodes appear together in sentences in the cluster. As explained before, they do not have to appear right next to each other in the sentence, they can be up to a $MINIMUM\_GAP$ words apart. The redundancy check also helps with the performance of the traversal and ensures that only the redundant paths are followed. It is specified in the form of a $MINIMUM\_REDUNDANCY$ parameter which is empirically set in the basic Opinosis Graph to 2. We modify this parameter to 3 or higher depending on the density of the graph and the number of comments in the thread. We dynamically modify the parameter based on the number of occurrences of each query word as such.

$$MINIMUM\_REDUNDANCY = \begin{cases} 1 & if(QUERY\_FREQUENCY < 15) \\ 2 & if(15 \leq QUERY\_FREQUENCY \geq 25) \\ 3 & if(QUERY\_FREQUENCY > 25) \end{cases} \cdot \tag{3.9}$$

Algorithm 3 shows our algorithm we created for traversing the paths in SPORK. This algorithm is given as input the starting query node and the direction of traversal. If the direction is *FORWARD* it will propagate using the outgoing edges of the nodes, and vice versa for the *BACKWARD* direction. This algorithm is called using the query word as the starting node and all forward and backward propagations are stored as shown in Algorithm 4.

These two separate lists of partial paths are then stitched together to create potentially valid sentences. At this point, the paths are checked for validity. The Opinosis team suggested using a set of regular expressions that contained part of

Algorithm 3 Graph Propagation

Parameters: $MIN\_REDUNDANCY = 3$

Output: list of propagations

procedure PROPAGATE($query, direction$)

$v_k \leftarrow last(list)$

if $RedundancyCheck(list_{k-1}, v_k) \geq MIN\_REDUNDANCY$ **then**

if $direction == FORWARD$ **then**

if $VEN(v_k)$ **then**

$AddForwardPropagation(list)$

end if

else

if $VSN(v_k)$ **then**

$list \leftarrow Reverse(list)$ ▷ Ensure proper ordering for path.

$AddBackwardPropagation(list)$

end if

end if

for all $v_n \in Edges(v_k, direction)$ **do**

$list.add(v_n)$

Traverse($list$)

end for

end if

end procedure

Algorithm 4 Graph Traversal

Output: list of valid paths

```
procedure TraversePaths(queries)
    for all query ∈ queries do
        forwardPropagations ← Propagate(query, FORWARDS)
        backwardPropagations ← Propagate(query, BACKWARDS)

                    ▷ Stitched paths are created by simply joining every forward
propagation with every backward propagation.
        potentialPaths ← (stitchPaths(forwardPropagations, backwardPropagations))
        for all potentialPath ∈ potentialPaths do
            if isValidPath(potentialPath) then
                validPaths.append(potentialPath)
            end if
        end for
    end for
    return validPaths
end procedure
```

speech patterns that the sentences were required follow. If a path was found that did not match a known POS pattern it was thrown out. This solution increases the comprehensiveness of the sentences but does not guarantee a grammatically correct sentence. Additionally, following a set of rules limits the portability of the sentence. Certain sentence types can be created using this approach such as comparative sentences but, they do not encompass the entirety of a summary. Although several regular expressions were tested, they proved to be of limited usefulness as the sentences were not guaranteed to be well-formed and the performance of SPORK was compromised. Instead, simply testing for appropriate sentence length and requiring the sentences contain a *verb* yielded the highest ranking sentences. This simpler check proved to be more efficient and still contain all high ranking sentences. Though the generated structures are not always well-formed sentences, they contain useful information that will help us in the extraction step in Section sentence-extraction.

Example. Although the next section explains the process of scoring and ranking the sentence paths, the traversed paths are broken down here. In Sections 3.3.3, 3.4.1, and 3.4.2 the clusters are created, queries are generated, and graph is created. As explained in this section, the queries are used to run the traversal on each graph cluster. Using the top 3 clusters from before, Table 3.6 shows how many sentences are generated by each query.

Cluster	Query Term	Sentences Generated
1	performance	165,016
1	power	124,168
4	thing	209,715
4	core	204,847
5	cores	647,745
5	people	116,366

Table 3.6: *Graph Traversal Example Output.* This table shows the stats of the output of the traversed paths of the 3 largest graph clusters created in Section 3.4.2.

3.4.4 Path Scoring

Once all of the paths are generated, a scoring algorithm is run to rank and order important paths. Several different scoring algorithms were tested including the ones provided in the Opinosis Graph. Three basic scoring algorithms were suggested in the Opinosis Paper and are explained below [17].

- **Simple Scoring:** The simple scoring algorithm is defined in Equation 3.10, where the score of path W is calculated. It counts the number of redundancies between two nodes based off of the meta information between both of them and the $MINIMUM\_GAP$ parameter. This represented as the $r()$ function. The total path score is then counted as the sum of all of the edge scores divided by the total number of words in the path, which is represented as $|W|$. While the simple scoring algorithm simply averages the edge scores it does not give any kind of bias towards longer sentences. This usually results in the simple algorithm returning sentences that are

short but have high edge redundancies.

$$S_{basic}(W) = \frac{1}{|W|} \sum_{k=i+1,i}^{s} r(i,k) \tag{3.10}$$

- **Incremental Scoring:** The incremental scoring algorithm takes into account the length of the path and is shown in Equation 3.11. v_i is the first node in the path being score and v_s is the last node. The algorithm scores each edge by multiplying the edge redundancy by the position the word is in the sentence. $|v_i, v_k|$ represents the length of edges in between the current node v_k and the beginning node v_i. These scores are then summed up and divided by the number of words in the path. This means that words that are later in the sentence will get higher scores and therefore the paths should not be biased towards short sentences. This technique instead usually biases towards longer sentences and can produce the opposite result of the first algorithm.

$$S_{wt\_len}(W) = \frac{1}{|W|} \sum_{k=i+1,i}^{s} |v_i, v_k| * r(i,k) \tag{3.11}$$

- **Normalized Length:** This last scoring algorithm shown in Equation 3.12. It similarly weights the score by the path length, however it weights it in a way such that the length of the path does not dominate the score. This equation is most similar to the incremental scoring, however, it scales down the path length by taking the log of it.

$$S_{wt\_loglen}(W) = \frac{1}{|W|}(r(i,i+1) + \sum_{k=i+2,i+1}^{s} \log_2|v_i, v_k| * r(i,k)) \tag{3.12}$$

The normalized length approach proved to be the most effective; however, it was still lacking in a couple areas. Although most redundant paths were followed, paths that were composed of several common words, were created. To account

for this the TF-IDF measure calculated in Section 3.3.2 was used to balance out the less important words. The TF-IDF score accurately measures which words are significant specifically to the given post and enables the words to be weighted to create better paths.

An additional step to the path score was using information given by the structure of the Reddit comment thread. Comments are themselves weighted by a score determined by the users. Each comment is up- or down-voted, which can help determine the popularity or importance of a given comment. The score can help determine the importance of a given comment and by extension the sentences and words in the comment however, it should not be the sole scoring mechanism. Comments can be up-voted for reasons other than significance to the document, for example if the comment contains humor. However, when used in conjunction with other scoring mechanisms it proves to be effective. Since nodes contain many instances of words, each node is given a total up-vote score which is a sum of all of the up-vote scores of the individual meta-informations.

The final scoring formula is calculated as such:

$$S_{final}(W) = \frac{1}{|W|}\Bigg(r(i,i+1) + \sum_{k=i+2,i+1}^{s}\Bigg(\log_2|v_i,v_k| * r(i,k) * tfidfScore(v_k) * UpVoteScore(v_k)\Bigg)\Bigg) \tag{3.13}$$

Overall, the scoring method uses the normalized length algorithm introduced in the Opinosis Graph and was extended to account for the additional features

provided in this graph.

Example. This is the second-to-last step in the running example. It demonstrates what the traversed paths from Section 3.4.3 look like. Table 3.7 shows the highest ranking path structures according to the final scoring algorithm defined in this section. Although the paths do not make coherent sentences, they contain the strongest connections between words from sentences in the comment clusters. Not only that, but they capture sequenced gaps and take into account the crowd-sourced comment scoring implemented on Reddit. They also weigh their scores based on stop-word rankings. Additionally, they follow the valid sentence rules outlined in Section 3.4.3. Although more rigid rules can be defined to create better sounding sentences, there is no set of rules that can define all types of valid sentences, especially ones that contain slang, proper nouns, outside references, and incorrect words common on Reddit. Instead of trying to achieve this, the sentence structures with the highest redundancies and most important identifying information are used to extract real sentences out of the original cluster as shown in the final section, 3.4.5.

Cluster	Query Term	Top Ranked Path Structure
1	performance	[“and”, “six”, “cores”, “to”, “compare”, “performance”, “.”]
1	power	[“and”, “six”, “cores”, “with”, “lower”, “power”, “it”, “’s”, “performance”, “.”]
4	thing	[“or”, “45ghz”, “of”, “the”, “bad”, “thing”, “is”, “this”, “cpu”, “.”]
4	core	[“or”, “45ghz”, “of”, “memory”, “per”, “core”, “is”, “this”, “cpu”, “.”]
5	cores	[“the”, “performance”, “of”, “cores”, “is”, “it”, “.”]
5	people	[“most”, “of”, “cores”, “is”, “that”, “people”, “to”, “the”, “cores”, “.”]

Table 3.7: *Ranked Paths, Example Output.* This table shows the highest ranking paths from all traversed paths generated in Section 3.4.3. The largest 3 clusters from the running example are used to generate the paths.

Something to note from these paths is that there is some redundancy between different queries within the same cluster. Although the queries help to target the paths, ultimately, the highest ranking paths are still found. This means that a lot of the time the highest ranking paths will either contain more than one of the highest ranking queries, or a slight variation in the path will pass through the different queries. This shows that clustering the comments properly is crucial to getting a good summary.

3.4.5 Sentence Extraction

Initially, the graph creation and graph traversal were the final steps of the pipeline and were used to create summary sentences. This setup is more like the one found in the basic Opinosis Graph and is ideal since it generates novel sentences not found in the text. This goes back to the idea of making an ab-

stractive summarization approach. However, after we tested and tweaked this method, we found that it was ineffective at producing summary sentences. The sentence paths generated were more often than not invalid sentences and sounded like gibberish, as seen in Table 3.7.

Although none of these sentence paths contain proper grammatical English sentences, they still contain valuable information that can be used. They contain nodes that have redundant and important links between them which were found using the ranking formula in Section 3.4.3. This caused us to change our approach and revert back to an extractive method. With an extractive method summary sentences are guaranteed to be at least as well-formed and grammatically correct as the authors who write them. In this approach, the sentence paths are related back to the original text. The sentence paths that are the most similar to the sentences from within the cluster of comments are chosen as the most salient and representative of a summary. This is the idea of sentence extraction step we present.

Sentence extraction is the final step in the pipeline and is necessary due to the inconsistent nature of the traversed paths. During sentence extraction each of the generated paths gets compared to every sentence in the cluster of comments. They are compared based on similarity (explained later) and we create a matrix for all of these comparisons. The idea is that if we find a highly ranked relevant sentence path it should still contain important information. The sentence from the text that is the most similar to that important sentence path should therefore also be important. After every sentence path gets compared to every sentence from the original cluster of comments, the most similar one is chosen.

As a quick overview, each cluster of comments contains a couple of queries.

Each of those queries generates several valid paths. Once those paths are scored and ranked, the top two highest scoring paths from each query is chosen. Those paths then get compared to sentences within the original text. The ones that are most similar are chosen as the summary sentence for each query within the cluster. This is achieved by using a number of different similarity measures described in the list below. This basically transforms the shallow abstractive summarization approach presented in the basic Opinosis Graph and turns it into an extractive approach where sentences from the original text are chosen as candidates for the summary.

Below is a list of all of the similarity measures used to compare the paths to sentences in the text. These are by no means the only sentence similarity measures, however, they are all well known methods. Any different similarity metric can be substituted in and considered future work.

1. **Jaccard Similarity**. The Jaccard similarity coefficient "is a similarity measure that compares the similarity between two feature sets [8]". It is defined as the size of the intersection of the words in the two sentences divided by the size of the union of the words in the two sentences. Equation 3.14 shows the Jaccard coefficient of two sets of words A and B.

$$Jacc(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.14}$$

2. **Dice Similarity**. The Dice coefficient, also known as the SrensenDice index is another metric used to measure word overlap in strings. Shown in Equation 3.15, the Dice coefficient is defined as twice the union of the words in both sentences divided by the sum of the number of words in both

sets.

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3.15}$$

3. **Cosine Similarity**. Cosine similarity is another well-known measure used to compare the similarity of sentences. It is used as the similarity measure for the hierarchical clustering in Section 3.3.3. As shown in Equation 3.16, the cosine similarity between two vectors of words is defined as the dot-product between the two vectors divided by the product of the magnitudes of both vectors. Essentially, cosine angle between the two vectors is being calculated.

$$similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{\parallel A \parallel * \parallel B \parallel} \tag{3.16}$$

4. **Minimum Edit Distance**. The minimum edit distance, also known as the Levenshtein distance, usually involves comparing strings. It calculates the minimum number of characters that need to be substituted, inserted, or deleted in order for the strings to be the same. Here, this method is applied instead to two lists of strings. It is calculated in a recursive bottom-up approach. The recurrence relation is identified as such:

$$lev_{A,B}(i, j) = \min \begin{cases} lev_{A,B}(i-1, j) + 1 \\ lev_{A,B}(i, j-1) + 1 \\ lev_{A,B}(i-1, j-1) + \begin{cases} 2 & if A_i \neq B_i \\ 0 & if A_i = B_i \end{cases} \end{cases} \tag{3.17}$$

This algorithm is run on each element in both lists. The first item in the minimum corresponds to a deletion, the second to an insertion, and the third to either a match or mismatch. The scoring bonus for a match or penalty for a mismatch can be altered depending on the algorithm. In the

case of the Levenshetin distance metric used here a match gets a score of 2 in addition with the previous element and the mismatch a 0 in addition with the previous element.

5. **Local Alignment**. While the minimum edit distance gets the smallest amount of edits in two sequences, local alignment, also known as the Smith-Waterman algorithm, finds the best alignment within two sequences. The algorithms are very similar and both use a recursive approach. While the local alignment algorithm was initially created to find local regions of similarity in DNA sequences it can be directly applied to find local regions of similarity between word sequences.

$$H_{i,j} = max \begin{cases} 0;\ i = 0\ or\ j = 0 \\ H_{i-1,j} + penalty1\text{Deletion} \\ H_{i,j-1} + penalty2\text{Insertion} \\ H_{i-1,j-1} + \begin{cases} penalty3 & if A_i \neq B_i\text{Mismatch} \\ bonus1 & if A_i = B_i\text{Match} \end{cases} \end{cases}$$

In this algorithm an $n \times n$ matrix is built with the recurrence relation showed in Equation **??**. This time, all gaps in alignment are given a penalty. The insertion, deletion, and mismatch penalties as well as the match bonus can range from implementation to implementation. In our implementation we use simple and straightforward scores for each as presented in Table 3.8.

Type	Name	Score
Deletion	$penalty1$	-1
Insertion	$penalty2$	-1
Mismatch	$penalty3$	-1
Match	$bonus1$	2

Table 3.8: *Local Alignment Scores.* This table shows the penalty and bonus scores for deletions, insertions, mismatches, and matches.

An insertion, deletion, and mismatch all get the simplest penalties of -1 while a match gets a bonus of $+2$. These values can be arbitrarily defined and more advanced methods can take advantage of similar terms. In other words, two similar words should yield something better than a complete mismatch. These advanced methods can be seen as future work for SPORK.

Either way, the local alignment algorithm is used to find the largest aligned subsequence in between the two string sequences. In our work, these string sequences are taken from full comments instead of simply sentences.

Example. This last step in the running example shows the extracted sentences based off of the generated paths in Section 3.4.4. All five measures from this section are employed to find sentences from the text. Table 3.9 shows the resulting extracted sentences from the generated paths in the largest cluster. For brevity, only paths from the largest cluster, cluster 5, are shown. Although some of the results are shown here, Chapter 4 will more closely analyze the performance of each similarity measure.

Query Term	**Top Ranked Path Structure**
cores	[“the”, “performance”, “of”, “cores”, “is”, “it”, “.”]
Similarity Measure	**Top Ranked Extracted Sentence**
Jaccard Similarity	“The shift is toward more cores in general.”
Dice Similarity	“The shift is toward more cores in general.”
Cosine Similarity	“What they are saying is that when you design your programs to take maximum advantage of all the cores, this is the kind of performance you can expect.”
Minimum Edit Distance	“Intel is coming.”
Local Alignment	“A single number can never characterize the performance of an architecture.”
Query Term	**Top Ranked Path Structure**
people	[“most”, “of”, “cores”, “is”, “that”, “people”, “to”, “the”, “cores”, “.”]
Similarity Measure	**Top Ranked Extracted Sentence**
Jaccard Similarity	“Most of them reason that it isn’t worth their time.”
Dice Similarity	“Most of them reason that it isn’t worth their time.”
Cosine Similarity	“What they are saying is that when you design your programs to take maximum advantage of all the cores, this is the kind of performance you can expect.”
Minimum Edit Distance	“So making the jump to *hundreds* of cores is not going to increase the performance by hundreds (or even tens in many cases).”
Local Alignment	“A single number can never characterize the performance of an architecture.”

Table 3.9: *Sentence Extraction Example Output.* This table shows the sentences extracted from the highest ranking paths calculated in Section 3.4.4. This table only displays results from the 2 top-ranked paths in comment cluster 5. It shows the output of each individual similarity measure explained in this section.

Some things to note from this table are that some of the similarity measures find the same sentence. This is most common with the Jaccard and Dice similarity measures. Another thing to note is that different queries in the same cluster can produce the same extracted sentences, as is the case with the minimum edit distance. This is due to the similarity that can occur between path structures from different queries. Chapter 4 explains the results of several different comment

threads run using the SPORK system in more detail.

Chapter 4

Results

As shown in Chapter 3, there are many steps involved in the pipeline of the SPORK system. From comment clustering to graph creation and sentence extraction, each of the algorithms plays an important role in the end goal of the system. This goal is aimed at reducing the information overload presented in the many blogs and discussion forums online. The target, in this case, is the news site Reddit, but it can ultimately be applied to any multi-document text. The goal is achieved by exploring different topics and building a graph that can discover important chains of words. These word chain structures are then used to find important sentences in the original text that are representative of important opinions or ideas. This chapter explains the process by which SPORK is tested and verified.

4.1 Data Gathering

First it is important to discuss the data used the testing approach. We use a number of Reddit posts retrieved from the subreddits */r/technology* and */r/politics*. Subsequently, all of the comments were retrieved from those posts and a snapshot of the site of each post was taken to ensure the correct number of comments would be present during testing. This is important because new comments are constantly being added, changing the outcome of the summary. Table 4.1 shows the ids of each post used in the evaluation along with the number of comments associated with each post. The particular posts were chosen strictly due to the number of comments they contained and the depth of the discussion. The posts all contain from 200 to 400 comments, a substantial amount, but still manageable for manual consumption. In addition to simply getting the posts and comments in question, several additional comment threads were required to train the keyword ranker. Section 3.3.2 explains the benefit of using additional information in the corpus to strengthen the results of the $tfidf$ keyword ranking measure. To that extent, hundreds of posts from both subreddits were pulled using Reddit's API, each containing hundreds of comments.

subreddit	post_id	comments	extracted clusters
/r/technology	1ejeu3	260	5
/r/technology	1edoot	229	4
/r/technology	1ee2m0	313	6
/r/technology	1e5p0c	325	6
/r/politics	1eebtm	327	5
/r/politics	1ech0y	366	6
/r/politics	1cm7c9	350	4

Table 4.1: *Reddit Posts Used For Analysis.* This table contains a list of all of the posts used during the testing step. Each post contains between 200 and 400 comments, ensuring solid cluster sizes.

4.2 Evaluation

In order to test the results of SPORK, a panel of 11 experts was assembled to obtain the "ground truth" from every cluster in the posts outlined in Table 4.1. The experts obtained these truths in the following manner.

1. The experts were split into groups of three. Each cluster was given two of the posts from the collected data.

2. The clustering algorithm was run on each post, resulting in several clusters per post. Only clusters that contained a meaningful number of comments, more than 20, for this evaluation were analyzed.

3. Each member individually read each cluster and wrote the major points essential to a summary. The instructions were explained as such. Each member generated a couple of "key phrases" which they deemed crucial to each cluster. The key phrases defined what the cluster was about and should have been narrowly focused on a specific idea or topic within the

thread. They should not have been in the form of a full sentence, but, instead, of partial phrases that contained key constructs and were essential in summarizing the cluster. They should not have been general or broad sweeping statements about the cluster that could encompass several things. They should have unraveled the "ground truth" of the cluster, as it were, and have been as objective as possible. Although many ideas might have been present in the cluster, the top 3 most important or prevalent key phrases should have been chosen individually.

4. After each member in the group finished a cluster, the group discussed and voted on the top three out of nine key points for each cluster.

Three people were assigned to each group to create some redundancy in exploring the key ideas in each cluster of comments. Initially, the members came up with their own ideas aided by the use of separate Google Spreadsheets. A fourth spreadsheet then accumulated the results from the individual members and allowed for voting of the top three. This ensured that multiple members agreed on the major topics present in the comment cluster.

This approach was run with all 11 experts, forming 3 groups of 3 and one remaining group of 2. Overall, all 7 posts were manually parsed for a total of 36 clusters containing between 20 and 75 comments each. This resulted in generating ground truths for every cluster of comments.

After all of the ground truths were obtained the SPORK system is separately run on the same posts. SPORK generated 2 sentences per cluster (since it uses the 2 query approach).

The results were collected and then compared against the established ground truths. Specifically, the sentences generated using the sentence extraction techniques in SPORK were compared with the key points discovered by the panel of experts. A meaningful sentence generated by SPORK is one that contains one or more of the ground truths. Although the different sentence extraction approaches were sometimes similar, overall they all produced slightly different results. Some aligned better with the expert ground truths than others.

4.3 Analysis

Running SPORK on all 7 posts resulted in extracted sentences for the 36 clusters. Using the top 2 query approach from Section 3.4.1, 2 sentences were generated by each sentence extraction technique. Each sentence was compared to the top ground truths generated by the panel of experts in the cluster. If either of the extracted sentences matched any of the top 3 expert points in the cluster, SPORK was considered to have found a key point. We call this a match. To demonstrate this, Table 4.2 shows a list of all of the expert opinions coming from the second cluster of post *1ech0y*. As explained, each of these are key points that have been voted upon by three experts and determined to be ground truths. Table 4.3 shows each of the sentence extraction measures used on the same cluster and weather or not the results capture any of the expert opinions.

#	Expert Opinion
1	"The argument is whether it [Bitcoin - SL] is a currency or commodity."
2	"Bitcoin could be better than USD, banks are dysfunctional and the adoption could help restructure."
3	"The US Government is opposed to Bitcoins because they are something that lies outside their sphere of influence, but is otherwise similar to USD."

Table 4.2: *Expert Opinions For Post* *1ech0y.* This table contains a list of the top 3 expert opinions agreed upon by the experts. This comes from the second cluster of post *1ech0y*

Similarity Measure	Captured Sentence	Expert Opinion
Jaccard	"Bitcoin is not a currency it is a commodity."	1
Jaccard	"Bitcoin is not a currency it is a commodity."	1
Dice	"I'm treating the currency as a commodity to be bought/sold."	1
Dice	"Bitcoin is not a currency it is a commodity."	1
Cosine	"There are many reasons governments don't like it and want to make it as much of a hassle to use as possible."	3
Cosine	"Bitcoin is not a currency it is a commodity."	1
Min. Edit Dist.	"I made a profit."	None
Min. Edit Dist.	"Bitcoin is not a currency it is a commodity."	1
Loc. Alignment	"You're treating a currency as a commodity, which just lends to the idea of its instability."	1
Loc. Alignment	"I'm treating the currency as a commodity to be bought/sold."	1

Table 4.3: *Expert Opinion Comparison.* This table shows which expert opinions the summarized sentences match. The extraction method uses the top 2 query approach explained in Section 3.4.1

The summarized sentences match the expert opinions if they are able to capture what the expert opinions say. For example, take the summarized sentence

"Bitcoin is not a currency it is a commodity.". The sentence does not contain the exact words of the expert opinion 1, because this would be extremely unlikely since the expert opinions are naturally generated and not taken from the cluster. Although they are not exact copies, the expert opinion captures the idea that several comments contain the specific argument of weather or not Bitcoin is a currency or commodity. The retrieved sentence addresses that exact idea. Interestingly, that is not the only sentence to capture that argument. The sentence "I'm treating the currency as a commodity to be bought/sold.", although explained in different words, also captures the same argument. This means there are multiple different sentences that can represent the expert opinion. This is only natural, as the expert opinions are aimed at summarizing what the several comments in the cluster say. Multiple matches to the same expert opinion or even to different expert opinions, as shown with the cosine similarity measure, can be found. The different similarity measures often capture the same expert opinions and even the different queries within the same similarity measure can capture the same expert opinion. Overall though, not every summarized sentence matches one of the expert opinions as is the case with the minimum edit distance measure.

These results come from just cluster 2 of post *1ech0y*. To see a detailed list of all of the matches see Appendix A.

To display the results for all of the clusters tested, Table 4.4 shows the how many of the expert opinions are captured in the summarized sentences. A match occurs with a similarity measure if either of the 2 query sentences matches any of the expert opinions. A match is represented with a "Y" for "yes" and a mismatch with "N" for "no". Since the results for each extraction method varies, a union

and majority match was calculated. If any of the extraction methods finds a match the union has a match. Similarly, if a majority of the extraction methods have matches, the majority has a match. The union and majority measures help us understand how well the different extraction methods within SPORK are working overall. A percent match is shown below each method providing an accuracy for all of the 36 clusters.

Most notably, the local alignment sentence extraction method performed the best, getting a match in 72% of the clusters tested. Each method performed better in some areas than others resulting in a high union match of 84%. Overall, SPORK performed admirably in discovering key topic points in the Reddit posts, excelling with the local alignment and cosine similarity measures. The Dice similarity, Jaccard similarity, and minimum edit distance measures on the other hand performed almost similarly poorly, maxing out with a match in only 36% of the clusters. The lower scores from these three measures resulted in a low majority match rate of 31%. There are several issues that were discovered while manually labeling data and while comparing results that held SPORK back.

These missteps can be traced back to several key points. First, during manual labeling, experts noticed some issues in the cohesiveness of the clusters. While some clusters genuinely appeared to revolve around similar ideas or topics, the experts noted that others appeared scattered and did not lead to anything conclusive. This caused issues when coming up with summary points due to the poor clustering of similar topics. Although this was not the case for all clusters, a couple clusters, marked with an "*" in Table 4.4, were noted by the experts as being non-cohesive. The root of this problem can be traced back to the clustering algorithm itself and the moderately ineffective surface level similarity measure.

post_id	Cluster	Jaccard	Dice	Cosine	Min. Edit	Loc. Align.	Union	Majority
1ejeu3	1	N	N	Y	N	Y	Y	N
1ejeu3	2	N	N	N	Y	Y	Y	N
1ejeu3	3*	N	N	N	N	N	N	N
1ejeu3	4	Y	N	Y	N	Y	Y	Y
1ejeu3	5	Y	Y	Y	N	Y	Y	Y
1eebtm	1	N	N	Y	Y	Y	Y	Y
1eebtm	2	N	N	Y	N	Y	Y	N
1eebtm	3	Y	Y	Y	N	N	Y	Y
1eebtm	4**	N	N	N	N	N	N	N
1eebtm	5	N	N	Y	N	Y	Y	N
1e5p0c	1	Y	N	Y	N	Y	Y	Y
1e5p0c	2	N	N	Y	N	Y	Y	N
1e5p0c	3	Y	N	N	N	N	Y	N
1e5p0c	4	Y	Y	Y	N	Y	Y	Y
1e5p0c	5**	N	N	N	N	N	N	N
1e5p0c	6 *	Y	Y	N	N	N	Y	N
1edoot	1	N	N	N	Y	Y	Y	N
1edoot	2	N	N	N	Y	Y	Y	N
1edoot	3	N	N	N	N	Y	Y	N
1edoot	4	N	N	N	N	N	N	N
1ech0y	1	N	N	N	N	Y	Y	N
1ech0y	2	Y	Y	Y	Y	Y	Y	Y
1ech0y	3	N	N	N	N	N	N	N
1ech0y	4	N	N	N	N	Y	Y	N
1ech0y	5	N	N	N	N	N	N	N
1ech0y	6*	N	N	N	N	Y	Y	N
1cm7c9	1	N	N	Y	N	Y	Y	N
1cm7c9	2	N	N	N	N	Y	Y	N
1cm7c9	3	N	N	Y	N	Y	Y	N
1cm7c9	4	Y	Y	Y	Y	Y	Y	Y
1ee2m0	1	Y	N	Y	N	Y	Y	Y
1ee2m0	2	Y	Y	N	N	N	Y	N
1ee2m0	3*	N	N	N	N	Y	Y	N
1ee2m0	4	Y	Y	Y	Y	Y	Y	Y
1ee2m0	5	N	N	N	N	Y	Y	N
1ee2m0	6	Y	Y	Y	Y	Y	Y	Y
Total	36	36%	25%	47%	22%	**72%**	83%	31%
Adj. Total	30	40%	27%	57%	27%	**80%**	90%	36%

Table 4.4: *Results of System on 36 Clusters.* This table contains the results of running the summarization system on all 36 manually labeled clusters.

A more in-depth semantic level of understanding is needed to better cluster like-comments to ensure cohesiveness.

In addition to having some questionably cohesive clusters, some clusters contain short non-important comments. Although the comments are correctly clustered, they are full of short miscellaneous sentences irrelevant to the summary. These clusters are marked with a "**" in Table 4.4. An example of these comments can be seen when looking at cluster 4 of the post with the id "1eebtm". Comments in this cluster contain phrases such as: "Thank you.", "Indeed.", "Owned.", "dude.", and so on. These *miscellaneous* clusters are simply the byproduct of the clustering algorithm and are automatically removed unless they contain more than 20 comments. When they are not removed, they prove problematic for not only the experts but also the summarization system. Important topics are hard to pinpoint and most information is irrelevant.

When we remove the non-cohesive and miscellaneous clusters that contained very limited value, the results of SPORK significantly increase. We call these the "Adjusted Results", represented in the last row of Table 4.4. Instead of measuring 36 clusters we now only compare 30 relevant clusters. We find that local alignment, again, performs the best, increasing by 8% to a total of 80%. This means that the local alignment technique was able to get a match with the expert ground truths in 80% of tested clusters. Although all match rates increased, none of them showed as much promise as local alignment. Cosine similarity came second with a 57% match rate.

In addition to improving the clustering technique, the similarity measures used for the sentence extraction have a significant impact on the results. Both the Jaccard and Dice measures are similar and rely strictly on word presence and

absence. This means they do not take into account the ordering of the words or structure of the sentence. The local alignment similarity measure is able to best take advantage of the information present in the traversed graph paths. The local alignment measure is able to find the longest subchain of similar words present in the path structure and compare it to the other sentence in the cluster. This is important because of the fact that the traversed paths do not contain full sentences. They contain partial fragments of connected words that show high redundancies. Therefore, it is less effective to compare the paths as if they were entire sentences then looking that the important structures within the path. When we consider that the local alignment technique only uses the simplest penalty and match scores and still manages to score the best, it becomes the most intriguing method. These advanced scoring methods for local alignment are explored as future work in Section 5.

Chapter 5

Conclusion and Future Work

Although there is room for improvement in the clustering, graph traversal, and sentence extraction algorithms, SPORK managed to perform well according to the expert opinions. In its best efforts, a 72% match of key topics in the tested clusters were found using the local alignment extraction algorithm and with cleaned up clusters an 80% match rate was achieved.

SPORK is a novel summarization pipeline that consists of three main steps: data collection, preprocessing, and summarization. These steps are split into 8 sub-steps that all cater towards building an original graph-based summarization approach. Although the Opinosis Graph [17] is used as the base for retrieving important sentence paths, it has been dramatically altered and adapted for the different types of content and goals the SPORK has.

First, data collection provides some interesting benefits and challenges. Since posts are categorized nicely into subreddits, using a metric, such as TF-IDF, that relies on multiple documents is very successful. The challenges spawn from the nature of online discussions. Online discussions vary greatly and diverge into

several topics and opinions. The goal of SPORK is to identify these individual topics and to find the key opinions within them.

In addition to the unique type of data used for summarization, SPORK attempts to cluster comments within the posts into coherent groups. These clusters are then much better suited for the graph-based approach taken in the summarization step. SPORK uses a cutting edge POS Tagger that yields accurate results for the graph.

SPORK also takes a unique approach on the basic Opinosis Graph. The graph creation has been adapted to incorporate the additional information by the structure of Reddit, such as comment scores. Additionally, to alleviate some of the complexity with larger graphs, graph traversal is now directed using a χ^2 query approach. This enables larger graphs to run with more accuracy and to discard some of the unnecessary traversals. The χ^2 ranking ensures that all query words are relevant throughout the cluster and the summary. Out of the top 10 ranked χ^2 terms, the ones with the highest occurrence in the cluster are used to ensure that a substantial amount of sentence paths are found. Traversals now start from the given query node and propagate in either direction. The forward and backward propagations are then stitched together to create the sentence paths. These sentence paths also use a new ranking approach that combines sentence scores and the TF-IDF ranking implemented earlier.

Finally, due to the inconsistent sentence paths, we devise a new solution that extracts sentences from the original clusters. We use the salient information provided in the sentence paths to get the most salient sentences from the text by using several different sentence similarity measures. We find the similarity of the sentence paths to the sentences of the original text and choose the most

similar ones as summary sentences. Although this transforms the approach into an extractive solution, it yields more positive results. The varying similarity measures that we use during the sentence extraction step has a huge effect on the results.

Some of the hurdles surrounding the graph generation and graph traversal algorithms are tied back to the original requirements of the Opinosis Graph only working on redundant opinions. These problems are alleviated by attempting to cluster comments into more focused groups, and furthermore, using targeted queries to limit the traversal of nodes in the graph. Not only are the graph generation and graph traversal algorithms fundamentally altered, the scoring approach is novel as well. The sentence path scoring algorithm benefits from the additional information of the corpus. Both using a TF-IDF keyword ranking approach, and using the scores provided by the Reddit voting system proved to return more meaningful sentence path structures. Since common words such as "the", "and", and "to" usually contain really high redundancies with other words, these words skew the score of the paths by weighting those higher. This means that before the inclusion of the keyword ranking technique, common words were much more frequent in the sentence paths. The actual value provided by using the paths was lowered as a result. Weighting words based off the corpus was effective at mitigating the problem and it ensured that more relevant structures appeared in the paths. All of these steps together combine to create a fairly effective pipeline for extracting summary sentences in a text with multiple opinions and ideas.

While these approaches benefit SPORK, there are a couple of areas in which improvements can be considered for future work. The improvements for future

work can be categorized as such: abstractive summarization, improved clustering, improved sentence extraction, and a friendly software wrapper.

While the main goals of automatic text summarization can be accomplished using extractive techniques, abstractive summarization is usually referred to when talking about a "true" summarization. Just as humans would read, parse, and generate a novel summary of some text, abstractive summarization generates new sentences not found in the original text. This type of summarization is highly desired and also extremely difficult. These difficulties come from a combination of creating a deep semantic understanding of the text and using natural language generation. An attempt at doing this is shown in the Opinosis Graph [17]. Although this is considered a "shallow" abstractive approach, it still generates novel sentences not found in the original text. These sentences are based on redundancies between words found in the graph. The problem with it comes during the traversal of the graph and the actual generation of new sentences. The algorithm used finds a sentence by connecting nodes starting with a VSN (valid start node) and ending with a VEN (valid end node). Additionally, we initially tested the generated sentence paths matched a certain set part-of-speech patterns as was done in the Opinosis Graph [17]. In theory this might sound good, but in practice, most paths that are created are gibberish sentences. This was a major hurdle in this research and motivated the need for sentence extraction approaches. Although the Opinosis Graph was tested on a much more limited corpus (manually labeled redundant reviews), it proved not to transfer over well to the more unstructured text used in this work. This leaves many possibilities for future work, including exploring different approaches in validating correct sentence paths. This would eliminate the need for the sentence extraction tech-

niques and transform this pipeline into abstractive, albeit shallowly abstractive, solution.

In addition to using better sentence validation approaches, ensuring that the sentences used contained more redundant opinions could help increase the comprehensiveness of the generated sentences as well. Although the very nature of web comments tends to lead to multiple, unstructured, sporadic conversations, clustering these more intelligently could reduce the problem. Since the Opinosis Graph used very redundant opinions successfully when generating sentences, having more focused and redundant comment clusters might be the key. Issues were found with the cohesiveness of the comment clusters as explained in Section 4, showing that the clustering technique can be improved. The suggestion leads to future work of using more intelligent and specifically semantic techniques for clustering comments. While the similarity measures used for clustering comments in Section 3.3.3 relied heavily on word presence and sentence structure, other more in-depth approaches rely on the semantics of the sentences. Work done in [8] and [19] outline new approaches to "utilize linguistic knowledge such as semantic relations between words and their syntactic composition, to determine the similarity of sentences [8]." Techniques such as word sense disambiguation are used to find the meaning of words in the sentences to find similarities. This is a much more in depth approach and could increase the effectiveness of the overall SPORK pipeline.

Another potential future improvement to SPORK could be the sentence extraction techniques in the last step of SPORK. Although there were 6 sentence extraction techniques tested, the local alignment formula scored the highest and showed the biggest potential. The easiest change in the local alignment formula

is to the penalties and bonuses of term deletions, insertions, mismatches, and matches as explained in Section 3.4.5. Although we used the simplest penalties and bonuses, each score can be approached differently. Take for example, a match and a mismatch. No matter how close a term is to another term, any difference between the two will lead to a mismatch. Instead, the match and mismatch scores could be determined based on how similar the two words are. If the terms are identical, the score could get a maximum bonus, while if the terms are completely opposite the score could get the maximum penalty. Wordnet is a useful tool that determines the similarity of words [14]. It contains a lexical database of English words that are grouped together into cognitive synonyms. Wordnet and other tools could help aid the process in scoring the penalties and bonuses provided in the local alignment similarity measure. Local alignment shows a lot of promise and improvements in it can potentially increase the effectiveness of SPORK.

Finally, as mentioned in the introduction (Section 1), a valuable extension of this work might include creating a more usable software wrapper. This wrapper could be in the form of a browser plugin and enable use of the SPORK tool in a more friendly or automated manner. This SPORK plugin could automatically run when visiting Reddit (and potentially other sites). When the SPORK plugin is run on a site it could potentially embed the summary sentences in a list above the actual comment thread. Not only does this make SPORK much more usable than a CLI tool but it helps illustrate the goals of this work. That is, to reduce the information overload on on interactive discussion-based sites.

These are just some of the many avenues for future work for the SPORK framework. Overall though, SPORK provides a novel framework for finding key opinions in multi-document web discussions.

Bibliography

[1] Tokenize package; NLTK 2.0 documentation. http://nltk.org/api/nltk.tokenize.html, 2012. NLTK - Natural Language Toolkit.

[2] API reddit/reddit wiki. http://www.reddit.com/, 2013. Github Inc.

[3] Python Reddit API Wrapper. https://github.com/praw-dev/praw, 2013. Github Inc.

[4] Reddit: the front page of the internet. http://www.reddit.com/, 2013. Reddit Inc.

[5] Reddit.com site info. http://www.alexa.com/siteinfo/reddit.com, 2013. Alexa Internet, Inc.

[6] Subreddit stats - stattit - reddit statistics. http://stattit.com/subreddits/, 2013.

[7] Wikipedia:statistics. http://en.wikipedia.org/wiki/Wikipedia:Statistics, May 2013. Wikimedia Foundation, Inc.

[8] P. Achananuparp, X. Hu, and X. Shen. The evaluation of sentence similarity measures. In *Data Warehousing and Knowledge Discovery*, pages 305–316. Springer, 2008.

[9] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[10] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[11] A. Celikyilmaz and D. Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 491–499, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[12] D. Das and A. F. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[13] G. Erkan and D. R. Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[14] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[15] K. Filippova. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[16] P. Fung, G. Ngai, and C.-S. Cheung. Combining optimal clustering and

Hidden Markov Models for extractive summarization. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering - Volume 12*, MultiSumQA '03, pages 21–28, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[17] K. Ganesan, C. Zhai, and J. Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 340–348, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[18] T. S. N. L. P. Group. Stanford pos tagger faq. http://nlp.stanford.edu/software/pos-tagger-faq.shtml.

[19] C. Ho, M. A. A. Murad, R. A. Kadir, and S. C. Doraisamy. Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 418–426. Association for Computational Linguistics, 2010.

[20] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 901–904, New York, NY, USA, 2007. ACM.

[21] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 291–298, New York, NY, USA, 2008. ACM.

[22] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[23] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32(4):485–525, Dec. 2006.

[24] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[25] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[26] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data*. Data-centric systems and applications. Springer-Verlag GmbH, 2007.

[27] F. Liu and Y. Liu. From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 261–264, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[28] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 131–140, New York, NY, USA, 2009. ACM.

[29] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large

annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330, 1993.

[30] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[31] K. McKeown and D. R. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 74–82, New York, NY, USA, 1995. ACM.

[32] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[33] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[34] S. Park and B. Cha. Query-based multi-document summarization using non-negative semantic feature and NMF clustering. In *Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on*, volume 2, pages 609–614, Sept.

[35] M. F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[36] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4):399–408, Dec. 2002.

[37] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[38] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[39] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[40] K.-F. Wong, M. Wu, and W. Li. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 985–992, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[41] X. Ye and H. Wei. Query-based summarization for search lists. In *Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on*, pages 330–333, Jan.

Appendix A

Results

This appendix contains all of the SPORK results run on the original 7 posts presented in Table 4.1. The *id* of the post is displayed at the top of the page and then each cluster, its subsequent expert opinions, and finally the SPORK results are displayed below it. Expert opinions that have a match with a SPORK summary sentence are highlighted. The different colors represent the different matches.

Post - t3_1ejeu3
Cluster 1
Article has a flawed premise. Not everyone needs 60k+ a year to get by
Healthcare is an expensive cost that is often no full appreciated in these calculations.
The location where you live greatly influences cost of living.
Queries - people, family
Cosine
* family:NN - 1200 x 12 for rent =14400(which is an absurd amount for poverty level in the majority of the country ....) 125 x 52 for food = 6500 200 x 12 for car = 2400 200 x 12 for second car = 2400 120 x 12 for insurance for 2 vehicles = 1440 300 x 12 for utilites water/electric/gas = 3600 400 x 12 for insurance fam of 4 = 4800 400 x 12 for misc spending clothes, gas etc =4800 federal taxes on 60k for joint filing with 2 defendants is only 2094 so 60k-(taxes)- expenses = 19660 slush fund for `` just getting by '' ... f *edit* some figures ya know i got ta say where the fuck does this 60k figure come from?
* people:NNS - 1200 x 12 for rent =14400(which is an absurd amount for poverty level in the majority of the country ....) 125 x 52 for food = 6500 200 x 12 for car = 2400 200 x 12 for second car = 2400 120 x 12 for insurance for 2 vehicles = 1440 300 x 12 for utilites water/electric/gas = 3600 400 x 12 for insurance fam of 4 = 4800 400 x 12 for misc spending clothes, gas etc =4800 federal taxes on 60k for joint filing with 2 defendants is only 2094 so 60k-(taxes)- expenses = 19660 slush fund for `` just getting by '' ... f *edit* some figures ya know i got ta say where the fuck does this 60k figure come from?
Jaccard
* family:NN - I 'm doing an analysis of how much tax this hypothetical family will pay.
* people:NNS - I 'm saying this is where i see the poverty level really being.
Dice
* family:NN - I 'm doing an analysis of how much tax this hypothetical family will pay.
* people:NNS - I 'm saying this is where i see the poverty level really being.
Min Edit Distance
* family:NN - With the exception of a major city like ny.
* people:NNS - But they are n't the same thing.
Local Alignment
* family:NN - 1200 x 12 for rent =14400(which is an absurd amount for poverty level in the majority of the country ....) 125 x 52 for food = 6500 200 x 12 for car = 2400 200 x 12 for second car = 2400 120 x 12 for insurance for 2 vehicles = 1440 300 x 12 for utilites water/electric/gas = 3600 400 x 12 for insurance fam of 4 = 4800 400 x 12 for misc spending clothes, gas etc =4800 federal taxes on 60k for joint filing with 2 defendants is only 2094 so 60k-(taxes)- expenses = 19660 slush fund for `` just getting by '' ... f *edit* some figures ya know i got ta say where the fuck does this 60k figure come from?
* people:NNS - 1200 x 12 for rent =14400(which is an absurd amount for poverty level in the majority of the country ....) 125 x 52 for food = 6500 200 x 12 for car = 2400 200 x 12 for second car = 2400 120 x 12 for insurance for 2 vehicles = 1440 300 x 12 for utilites water/electric/gas = 3600 400 x 12 for insurance fam of 4 = 4800 400 x 12 for misc spending clothes, gas etc =4800 federal taxes on 60k for joint filing with 2 defendants is only 2094 so 60k-(taxes)- expenses = 19660 slush fund for `` just getting by '' ... f *edit* some figures ya know i got ta say where the fuck does this 60k figure come from?
Cluster 2
$60k is the average because that's the average family income.
No matter what a family's income is, they always feel that they are "just getting by" because it is the lifestyle that they are used to.
Americans are out of touch with what real poverty is.
Queries - frugal, math
Cosine
* frugal:JJ - It 's that we are n't being frugal enough.
* math:NN - [here 's how to math it out.
Jaccard
* frugal:JJ - It 's that we are n't being frugal enough.
* math:NN - [here 's how to math it out.
Dice
* frugal:JJ - It 's that we are n't being frugal enough.
* math:NN - [here 's how to math it out.

Min Edit Distance
* frugal:JJ - $ 60k is the average because that 's the average family income.
* math:NN - [here 's how to math it out.
Local Alignment
* frugal:JJ - Whether you make $ 30k or $ 100k a year, you always feel like you are just getting by as your lifestyle / region adjust to your income.
* math:NN - Whether you make $ 30k or $ 100k a year, you always feel like you are just getting by as your lifestyle / region adjust to your income.
Cluster 3
A single federal poverty line is a bit silly anyway due to the variance of cost of living depending on where a person is located
Term limits and ineffective government policies make it difficult to properly address this problem.
Poverty is difficult to define because people's expectations vary.
Queries - lies, people
Cosine
* lies:VBZ - Lies!!
* people:NNS - An similar example would be the right to an attorney, where it is specifically provided that if you can not afford one, you will be appointed one by the state, because it is recognized that poor people should have the same opportunities as people with money.
Jaccard
* lies:VBZ - Lies!!
* people:NNS - It is not compatible with the near future.
Dice
* lies:VBZ - Lies!!
* people:NNS - It is not compatible with the near future.
Min Edit Distance
* lies:VBZ - Lies!!
* people:NNS - Reproduction is a fundamental human right.
Local Alignment
* lies:VBZ - Lies!!
* people:NNS - An similar example would be the right to an attorney, where it is specifically provided that if you can not afford one, you will be appointed one by the state, because it is recognized that poor people should have the same opportunities as people with money.
Cluster 4
Many people sharing statistics and crunching numbers to see if 24K is enough or 60K is actually poverty.
People argue that the cost of living is different in different places and cannot be equally compared across the board.
Many things affect the cost of living that many people do not consider when they think about estimates.
Queries - family, people
Cosine
* family:NN - Without kids, you would expect the family to owe $ 450.
* people:NNS - $ 60k is a decent amount of money for a small family in a lot of places.
Jaccard
* family:NN - Probably need to do that every week.
* people:NNS - $ 60k is a decent amount of money for a small family in a lot of places.
Dice
* family:NN - Probably need to do that every week.
* people:NNS - People are going to have sex.
Min Edit Distance
* family:NN - Probably need to do that every week.
* people:NNS - A lot of people do what my cousin does.
Local Alignment
* family:NN - I would certainly find having to share 60k on a family of 4 to be hard because i have expensive hobbies, but would i *need* 60k for basic needs?

* people:NNS - I would certainly find having to share 60k on a family of 4 to be hard because i have expensive hobbies, but would i *need* 60k for basic needs?
Cluster 5 - Expert Opinions
Living costs have been increasing drastically over time
Location is highly influential in housing costs
The cost of health care can vary greatly.
Queries - kids, struggle
Cosine
* kids:NNS - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.
* struggle:NN - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.
Jaccard
* kids:NNS - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.
* struggle:NN - I pay $ 410 for a one bedroom apt.
Dice
* kids:NNS - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.
* struggle:NN - I pay $ 410 for a one bedroom apt.
Min Edit Distance
* kids:NNS - Do you live in the us?
* struggle:NN - Would be a struggle with kids.
Local Alignment
* kids:NNS - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.
* struggle:NN - $ 24,000.00 in 1963 had the same buying power as $ 181,263.95 in 2013.

Post - t3_1eebtm
Cluster 1
People think that search should be just about search results and not extra JS or ads
Google search has been giving more vague answers and has been getting worse over the years.
Google search is improving in retrieving useful answers and has many advanced features.
Queries - google, results
Cosine
* results:NNS - I still think google needs to focus more on search than `` features "( see earlier gripe about showing the f1 results when i search `` f1 ", before i 've seen the race.
* google:NNP - I do n't see anything in google 's search page that is obtrusive and getting in the way of what i want.
Jaccard
* results:NNS - Google instead lists the results of the championship at the top of the search page.
* google:NNP - Google instead lists the results of the championship at the top of the search page.
Dice
* results:NNS - Time to sign out when i search.
* google:NNP - I just want to know what 's going on in the world.
Min Edit Distance
* results:NNS - I have nowhere to shop.
* google:NNP - My search experience w/ google has been going downhill.
Local Alignment
* results:NNS - Google instead lists the results of the championship at the top of the search page.
* google:NNP - They keep on adding more and more javascript and indirect links and frankly it gets in the way of me getting the info i want.
Cluster 2
People are confused about search engine directives and what does and doesn't apply.
People talk about how they don't know how to use all the features in Google Search.
People explain how to use some of the feature of Google Search.
Queries - search, work
Cosine
* search:NN - Now if you want to force a single word to be included, it has to be put in quotes.
* work:VB - Shortly after announcing google+, google stopped using the + sign to force a term to be used.
Jaccard
* search:NN - I know how a search engine is supposed to work.
* work:VB - I know how a search engine is supposed to work.
Dice
* search:NN - I know how a search engine is supposed to work.
* work:VB - I know how a search engine is supposed to work.
Min Edit Distance
* search:NN - Could be pretty awesome.
* work:VB - Could be pretty awesome.
Local Alignment
* search:NN - Shortly after announcing google+, google stopped using the + sign to force a term to be used.
* work:VB - If you 're logged in all the time then yeah but it 's also easier to hide that you 're looking at someone 's stuff if it 's on the google search page as opposed to them seeing you sifting through your gmail account.
Cluster 3
People discussed why the U.S. better quality products
People compare the new Google features to "New Coke".
People are considering moving to another product because of superfluous features and privacy.
Queries - google, permanently
Cosine
* permanently:RB - The mic will be permanently on for this to work.
* google:NNP - `` blind tests showed that people preferred `` new coke '' to old coke, but they rejected it out-of-hand nonetheless.
Jaccard
* permanently:RB - The mic will be permanently on for this to work.

* google:NNP - Is this going to be like the new coke?
Dice
* permanently:RB - The mic will be permanently on for this to work.
* google:NNP - Is this going to be like the new coke?
Min Edit Distance
* permanently:RB - Is google starting to jump the shark?
* google:NNP - `` written on the site.
Local Alignment
* permanently:RB - The mic will be permanently on for this to work.
* google:NNP - The preview text of the wiki article on the first page of google 's search for `` new coke '' is & gt; new coke was the reformulation of coca-cola introduced in 1985 by the coca-cola company to replace the original formula of its flagship soft drink, it was n't the 2000 's.
Cluster 4 - Bad Cluster: No data collected
Cluster 5
Advanced Search Features are hidden.
Google UX is bad.
People have privacy concerns with Google's products.
Queries - google, results
Cosine
* results:NNS - Many times, you search for `` term '', *in quotes*, it will still search for `` terms '' and `` termed '' and `` terming ''.
* google:NN - Specifically, i want to right click a link in google search results and copy the link.
Jaccard
* results:NNS - Specifically, i want to right click a link in google search results and copy the link.
* google:NN - Specifically, i want to right click a link in google search results and copy the link.
Dice
* results:NNS - Specifically, i want to right click a link in google search results and copy the link.
* google:NN - Specifically, i want to right click a link in google search results and copy the link.
Min Edit Distance
* results:NNS - I think he was being sarcastic and agreeing with you.
* google:NN - There 's been quite a bit of fuss over it.
Local Alignment
* results:NNS - Many times, you search for `` term '', *in quotes*, it will still search for `` terms '' and `` termed '' and `` terming ''.
* google:NN - Specifically, i want to right click a link in google search results and copy the link.

Post - t3_1e5p0c
Cluster 1
Anyone should be allowed to unlock their phones despite their carrier provider
You don't own the right of your cellphone until you pay for a contract
Congressmen don't know how technology works, therefore cant make good laws.
Queries - phone, contract
Cosine
* phone:NN - Seriously though, i think if you purchase the phone straight up, you should have the right to unlock the phone.
* contract:NN - These people selling unlock codes and tutorials on how to unlock your phone for you.
Jaccard
* phone:NN - Seriously though, i think if you purchase the phone straight up, you should have the right to unlock the phone.
* contract:NN - Unlock it.
Dice
* phone:NN - If the phone is off-contract then you can unlock it anyway.
* contract:NN - Unlock it.
Min Edit Distance
* phone:NN - Unlock it.
* contract:NN - Unlock it.
Local Alignment
* phone:NN - Seriously though, i think if you purchase the phone straight up, you should have the right to unlock the phone.
* contract:NN - These people selling unlock codes and tutorials on how to unlock your phone for you.
Cluster 2
Those that understand how to unlock a phone can already do so and those that don't will not.
The bill will not pass because the telecom companies will lobby against it
Congress is using this to get money from lobbyists
Queries - locked, session
Cosine
* locked:VBN - No doubt the telco lobby is buttering up the rest right now to vote against this bill.
* session:NN - No doubt the telco lobby is buttering up the rest right now to vote against this bill.
Jaccard
* locked:VBN - We are friends with the existing companies.
* session:NN - No doubt the telco lobby is buttering up the rest right now to vote against this bill.
Dice
* locked:VBN - We are friends with the existing companies.
* session:NN - No doubt the telco lobby is buttering up the rest right now to vote against this bill.
Min Edit Distance
* locked:VBN - There might not be enough time.
* session:NN - You already can.
Local Alignment
* locked:VBN - Heck, you do n't even need to know *what* a smartphone is, dumbphones can be locked too.
* session:NN - No doubt the telco lobby is buttering up the rest right now to vote against this bill.
Cluster 3
The people making the laws in the past didn't know much about technology. This law is an attempt to fix their old mistakes.
The government is attempting to take control of the air space (radio spectrum) and let telecom companies pay for a portion of the limited space.
Currently no locked cell phones are capable of being activated on a competing telecom network
Queries - congress, dmca
Cosine
* congress:NNP - Built into the dmca is the responsibility for the [library of congress to periodically review and decide which types of `` digital locks '' should be exempt from the anit-circumvention provision.

* dmca:NNP - Built into the dmca is the responsibility for the [library of congress to periodically review and decide which types of `` digital locks '' should be exempt from the anit-circumvention provision.
Jaccard
* congress:NNP - How noble of congress.
* dmca:NNP - I believe this proposed act is intended to remedy the problems with the anti-circumvention provision of the dmca.
Dice
* congress:NNP - How noble of congress.
* dmca:NNP - How noble of congress.
Min Edit Distance
* congress:NNP - How noble of congress.
* dmca:NNP - People argued this before the dmca.
Local Alignment
* congress:NNP - Built into the dmca is the responsibility for the [library of congress to periodically review and decide which types of `` digital locks '' should be exempt from the anit-circumvention provision.
* dmca:NNP - Built into the dmca is the responsibility for the [library of congress to periodically review and decide which types of `` digital locks '' should be exempt from the anit-circumvention provision.
Cluster 4
Everyone acts the Congressmen are stupid but they most likely know how to research and are somewhat intellegent. However they aren't acting of the peoples behalf.
Other countries have better phone deals
No one really understands how phone locking works.
Queries - phone, people
Cosine
* phone:NN - Since my return to the us from a bunch of time spent there, i 've stuck to buying unlocked phones from overseas because i do n't want to deal with it all.
* people:NNS - Yeah, i 'd never buy one of those `` unlocked '' phones since it 's totally vulnerable to search-and-seizure.
Jaccard
*** phone:NN - What does `` unlock '' mean in regards to a cellphone?**
* people:NNS - Yeah, i 'd never buy one of those `` unlocked '' phones since it 's totally vulnerable to search-and-seizure.
Dice
*** phone:NN - What does `` unlock '' mean in regards to a cellphone?**
* people:NNS - Yeah, i 'd never buy one of those `` unlocked '' phones since it 's totally vulnerable to search-and-seizure.
Min Edit Distance
* phone:NN - **we** want to be able to unlock our cellphones.
* people:NNS - I 'm open to enlightenment.
Local Alignment
* phone:NN - They made it *illegal* to do this not too long ago and now, all of a sudden, they want to legalize it?
*** people:NNS - I recall someone using the phrase `` everyone is selectively retarded, '' and it 's an apt statement here-- every single politician in world history has said something incredibly stupid, they 're not any different from other people.**
Cluster 5
The comments are not really related to the articles topic.
Sprint has good service but people give it a bad name
People who didnt read the article found the title confusing
Queries - people, good
Cosine
* good:JJ - Dmca has good stuff in it.
* people:NNS - The problem is that one single vote is meaningless but thousands of single votes are not.
Jaccard
* good:JJ - Dmca has good stuff in it.
* people:NNS - The problem is that one single vote is meaningless but thousands of single votes are not.

Dice
* good:JJ - Dmca has good stuff in it.
* people:NNS - The problem is that one single vote is meaningless but thousands of single votes are not.
Min Edit Distance
* good:JJ - How kind of them.
* people:NNS - How kind of them.
Local Alignment
* good:JJ - This does n't have a snowball 's chance in hell of passing.
* people:NNS - This does n't have a snowball 's chance in hell of passing.
Cluster 6
If you paid your contract you can get your provider to unlock your phone.
People don't understand what unlocking a cellphone true is.
The price for cellular services in the United States is higher than anywhere else in the world
Queries - phone, unlocking
Cosine
* unlocking:NN - Do you know what unlocking a phone means?
* phone:NN - ``
Jaccard
* unlocking:NN - Do you know what unlocking a phone means?
* phone:NN - Who 's your provider?
Dice
* unlocking:NN - Do you know what unlocking a phone means?
* phone:NN - Who 's your provider?
Min Edit Distance
* unlocking:NN - Bro do you even phone?
* phone:NN - Bro do you even phone?
Local Alignment
* unlocking:NN - How do you know when somebody does n't own a cell phone? ...
* phone:NN - How do you know when somebody does n't own a cell phone? ...

Post - t3_1edoot
Cluster 1
It is dangerous to short the yellow light time by shortening the time for drivers to react.
Police departments should not be funded on citations because it is a conflict of interest
Florida sucks
Queries - state, florida
Cosine
* florida:NNP - The concentration of elderly on the coasts and the backwoods weirdos in the central state make it pretty ridiculous.
* state:NN - The concentration of elderly on the coasts and the backwoods weirdos in the central state make it pretty ridiculous.
Jaccard
* florida:NNP - The difference in florida is that there is a higher sales tax and a higher tax on home ownership.
* state:NN - Those two *mostly* make up for the no state income tax.
Dice
* florida:NNP - The difference in florida is that there is a higher sales tax and a higher tax on home ownership.
* state:NN - Those two *mostly* make up for the no state income tax.
Min Edit Distance
* florida:NNP - This is dangerous.
* state:NN - That 's actually a thing.
Local Alignment
* florida:NNP - People call it a `` standing green light '' or `` stale green light ''- it refers to the situation where you can see the light for a significant amount of time before you get there; if it 's green the whole time, it may behoove you to assume it 's going to turn yellow when you 're on top of it and get out of the gas before you 're on top of it, so you 're more prepared to stop.
* state:NN - People call it a `` standing green light '' or `` stale green light ''- it refers to the situation where you can see the light for a significant amount of time before you get there; if it 's green the whole time, it may behoove you to assume it 's going to turn yellow when you 're on top of it and get out of the gas before you 're on top of it, so you 're more prepared to stop.
Cluster 2
Flordia has no way for a revenue stream.
The goal of traffic cameras is to increase profits for the state that they are in not for safety
The governor of florida is insane
Queries - state, describe
Cosine
* describe:VB - It is biased.
* state:NN - I 've always said that florida is the worst state in america.
Jaccard
* describe:VB - It is biased.
* state:NN - I 've always said that florida is the worst state in america.
Dice
* describe:VB - It is biased.
* state:NN - I 've always said that florida is the worst state in america.
Min Edit Distance
* describe:VB - Public safety is a scapegoat.
* state:NN - Not at all the worst state.
Local Alignment
* describe:VB - If y'all have n't figured it out yet the bottom line in any enforcement, deep down is for revenue.
* state:NN - I 've always said that florida is the worst state in america.
Cluster 3
The cameras at intersection aren't for public safety but rather a scam to gather cash.
Increasing the yellow light times but adding speed cameras would be a better option
Pedestrian walk signals can help warn you before a light changes
Queries - light, yellow
Cosine

* light:NN - Also, yellow lights are meant to allow traffic to clear out and people to stop.
* yellow:JJ - Also, yellow lights are meant to allow traffic to clear out and people to stop.
Jaccard
* light:NN - Now to yellow and red lights.
* yellow:JJ - Now to yellow and red lights.
Dice
* light:NN - Now to yellow and red lights.
* yellow:JJ - Now to yellow and red lights.
Min Edit Distance
* light:NN - Now to yellow and red lights.
* yellow:JJ - Screw.
Local Alignment
* light:NN - Something like every 10mph needed xx number of seconds for the light to stay yellow.
* yellow:JJ - I suppose the best option would be to just have speed cameras instead, while lengthening the yellow light.
Cluster 4
Flordia is a wonderful place to live and people don't know what they are talking about.
The length of the yellow light has to give sufficient stopping time
Bad traffic light setups cost money and gas
Queries - florida, traffic
Cosine
* florida:NNP - The investigations proved that the timers were shortened at only the lights with cameras.
* traffic:NN - There were lawsuits a few years back when these cameras were first enacted when this exact same thing happened.
Jaccard
* florida:NNP - The florida lights were reduced by .2 seconds.
* traffic:NN - The arguments for traffic laws are indeed quite sound.
Dice
* florida:NNP - The florida lights were reduced by .2 seconds.
* traffic:NN - The arguments for traffic laws are indeed quite sound.
Min Edit Distance
* florida:NNP - Good way to stimulate the economy.
* traffic:NN - Civil engineer, although not traffic.
Local Alignment
* florida:NNP - The cameras were just turned on within the last year in jacksonville when it was deemed to no longer be a questionable practice.
* traffic:NN - There were lawsuits a few years back when these cameras were first enacted when this exact same thing happened.

Post - t3_1ech0y
Cluster 1
Money laundering and tax evasion are FBI things. Likens bitcoins to internet gambling in the United States questioning the effectiveness of our government.
Bitcoins are not under control of the feds and are incredibly unstable, deterring their use.
The Federal siezure of money from MtGox doesn't actually affect the exchange of Bitcoins.
Queries - market, money
Cosine
* market:NN - What does this mean for the currency itself?
* money:NN - You can get around it by being able to exchange bitcoins for some other currency and then using the other currency to $ exchange rate to estimate value but you still have n't necessarily realized the gains.
Jaccard
* market:NN - When enough people are doing the quick transactions the market will even out enough for some stability.
* money:NN - `` the point of the post was to state that mtgox was trustworthy with my money.
Dice
* market:NN - Dwolla is/was the easiest way for us citizens to buy bitcoins.
* money:NN - Homeland thieves acting on behalf of the federal government to prevent currency competition and economic freedom.
Min Edit Distance
* market:NN - What does this mean for the currency itself?
* money:NN - The alternative is much worst.
Local Alignment
* market:NN - Bit-coins unlike currencies back by governments have built in deflation built into it which is worse because as more people use them for trade their value increases continuing resulting in hyper deflation which we have been seeing in the currency for the past year and makes the currency very unstable to use as a means of trade.
* money:NN - Homeland thieves acting on behalf of the federal government to prevent currency competition and economic freedom.
Cluster 2
Bitcoin is good as a means of transfering money or business transactions. The arguement is whether is is a currency or commodity.
Bitcoin could be better than USD, banks are disfunctional and the adoption could help restructure.
The US Government is opposed to biitcoins because they are something that lies outside their sphere of influence, but is otherwise similar to USD.
Queries - currency, reason
Cosine
* currency:NN - There are many reasons governments do n't like it and want to make it as much of a hassle to use as possible.
* reason:NN - Bitcoin is not a currency it is a commodity.
Jaccard
* currency:NN - Bitcoin is not a currency it is a commodity.
* reason:NN - Bitcoin is not a currency it is a commodity.
Dice
* currency:NN - I 'm treating the currency as a commodity to be bought/sold.
* reason:NN - Bitcoin is not a currency it is a commodity.
Min Edit Distance
* currency:NN - I made a profit.
* reason:NN - Bitcoin is not a currency it is a commodity.
Local Alignment
* currency:NN - You 're treating a currency as a commodity, which just lends to the idea of its instability.
* reason:NN - I 'm treating the currency as a commodity to be bought/sold.
Cluster 3
Bitcoins wont work because companies wont accept them.
Bitcoins are better as an investment tool than as a currency.

Money laundering and connections to the drug trade are problems shared by both bitcoins and by actual currencies.
Queries - currency, money
Cosine
* currency:NN - You mean the currency invented by printing press manufactures?
* money:NN - Did ... did you just compare a currency to a money transfer service?
Jaccard
* currency:NN - Did ... did you just compare a currency to a money transfer service?
* money:NN - Did ... did you just compare a currency to a money transfer service?
Dice
* currency:NN - Did ... did you just compare a currency to a money transfer service?
* money:NN - Did ... did you just compare a currency to a money transfer service?
Min Edit Distance
* currency:NN - Quick in and out?
* money:NN - You mean the currency invented by printing press manufactures?
Local Alignment
* currency:NN - Guarantee you he missed out on the 1000 % rise where he could have made some money and is now extremely butthurt because others are doing their thing so he has to try to shit on their parade.
* money:NN - Did ... did you just compare a currency to a money transfer service?
Cluster 4
The FED took money from Mtgox, bitcoin was uneffected.
Central banks help people by making the currency fluctuate less.
Bitcoins avoid banks and credit comapnies, but the action they would have gotten just goes to the exchange client like MtGox instead.
Queries - money, bitcoins
Cosine
* bitcoins:NNS - They are money.
* money:NN - It 's included in the blockchain.
Jaccard
* bitcoins:NNS - They are money.
* money:NN - It 's included in the blockchain.
Dice
* bitcoins:NNS - They are money.
* money:NN - It 's included in the blockchain.
Min Edit Distance
* bitcoins:NNS - They are the digital equivalent to a dollar note.
* money:NN - Mtgox is kind of shady.
Local Alignment
* bitcoins:NNS - They have a lot of money in bitcoins, but since they work as a currency exchange, they also had money in usd.
* money:NN - It cuts out credit cards and banks from the action but lets in mt.
Cluster 5
Smart people investing is not a sign of its quality, mainstream investors do not invest in it.
Bitcoins have value insofar as people say it has value. If people lose faith, bitcoins will fail.
Government services are apparently forced on us for the benefit of the corrupt underprivileged, mafia style.
Queries - people, quoted
Cosine
* quoted:VBD - You even quoted that.
* people:NNS - The thing that needs to be realized is that the law, in this case, is created by mathematical rules influencing supply rather then governmental policy.
Jaccard
* quoted:VBD - You even quoted that.
* people:NNS - Yeah, the funny part is that they 're not even rare anymore.
Dice
* quoted:VBD - You even quoted that.
* people:NNS - Yeah, the funny part is that they 're not even rare anymore.

Min Edit Distance
* quoted:VBD - You even quoted that.
* people:NNS - You even quoted that.
Local Alignment
* quoted:VBD - You even quoted that.
* people:NNS - I did n't, say, call that pirate was an obvious scam along with the *head developer* of the original bitcoin project, with an announcement put up as a banner on every page of the biggest bitcoin community, well before pirate started to default.
Cluster 6
Bitcoin's value is purely based on peoples confidence in it.
Bitcoins can't be siezed and can be useful in exchange provided you have someone willing to trade you USD for them.
Bitcoin can't really buy anything directly, but it can be used to buy gift cards that can then be exchanged for goods and services.
Queries - hard, bitcoin
Cosine
* hard:JJ - Interested to hear why you hold bitcoin in such high esteem.
* bitcoin:NNP - Bitcoin is n't the internet.
Jaccard
* hard:JJ - Interested to hear why you hold bitcoin in such high esteem.
* bitcoin:NNP - Bitcoin is n't the internet.
Dice
* hard:JJ - Interested to hear why you hold bitcoin in such high esteem.
* bitcoin:NNP - Bitcoin is n't the internet.
Min Edit Distance
* hard:JJ - Then you win.
* bitcoin:NNP - Bitcoin is n't the internet.
Local Alignment
* hard:JJ - Interested to hear why you hold bitcoin in such high esteem.
* bitcoin:NNP - While you ca n't buy it directly you can buy gift cards from gyft using bitcoin and then use them to buy gas and food.

Post - t3_1cm7c9

Cluster 1

People should be more aware of how corrupt the government is!

CISPA is not as far reaching as some believe, it is more to protect networks and make it possible to report crimes.

CISPA will allow the government to acess private information

Queries - information, cispa

Cosine

* information:NN - The implications of the government knowing everything about you is even bleaker, especially considering the dark road the usa is heading in regarding civil rights.

* cispa:NNP - The same is the case with cispa.

Jaccard

* information:NN - Only information 'directly pertaining to a vulnerability of, or threat to a system or network of a government or private entity, including information pertaining to the protection of a system or network' is included.

* cispa:NNP - The same is the case with cispa.

Dice

* information:NN - The eff itself is a lobbying group with agendas to push.

* cispa:NNP - The same is the case with cispa.

Min Edit Distance

* information:NN - The same is the case with cispa.

* cispa:NNP - This is incredibly intellectually dishonest.

Local Alignment

* information:NN - A system like the copyright blocking on youtube etc seems more likely where the government is essentially handed back-door access to the databases of all major internet companies.

* cispa:NNP - In fact, it encourages the sharing of all information with the government.

Cluster 2

Contact your rep!

CISPA will take away your internet privacy

America is in danger of losing it's freedom and becoming totalitarian

Queries - data, love

Cosine

* love:VBP - You hand it over.

* data:NNS - You 're important enough that the government wants your data.

Jaccard

* love:VBP - You hand it over.

* data:NNS - You 're important enough that the government wants your data.

Dice

* love:VBP - You hand it over.

* data:NNS - You 're important enough that the government wants your data.

Min Edit Distance

* love:VBP - You hand it over.

* data:NNS - The police desperately wants all your data 2.

Local Alignment

* love:VBP - I made so many calls that i actually went over my minutes.

* data:NNS - Companies have nothing to lose from giving your data to the government 4.

Cluster 3

CISPA will allow private companies to give your information to the federal government without the government having to get a warrant.

Personal info is okay to give out in criminal situations

All the government officials care about is how they can get more money.

Queries - data, information

Cosine

* information:NN - It is up to the government to use that info responsibly, which while i agree they probably wo n't, this does n't mean that companies should n't be allowed to give the government information if they want.

* data:NNS - Requiring a warrant to look at data that was volunteered to you is absurd.

Jaccard

* information:NN - And they share that information to the tech community as a whole.
* data:NNS - Requiring a warrant to look at data that was volunteered to you is absurd.
Dice
* information:NN - And they share that information to the tech community as a whole.
* data:NNS - Requiring a warrant to look at data that was volunteered to you is absurd.
Min Edit Distance
* information:NN - Dear non-new england representatives, wake the fuck up.
* data:NNS - Needing a warrant for that is absurd.
Local Alignment
* information:NN - It is up to the government to use that info responsibly, which while i agree they probably wo n't, this does n't mean that companies should n't be allowed to give the government information if they want.
* data:NNS - `` yes, so the government does n't consider your comment to be a crime- but now they have your data, and there is no provision to have non-relevant data deleted- so it sits there, forever, and since the government does n't need a warrant to view data it was handed by a company, if the political winds change, that data may come back to haunt you.
Cluster 4
Calls to congressmen do not really change policy
Not enough of the people who are complaining have actually read the bill.
CISPA exists in order to get a more accurate, well-rounded depiction of cyber-threats.
Queries - read, crossed
Cosine
* read:VB - Have you read the bill?
* crossed:VBD - Have you read the bill?
Jaccard
* read:VB - Have you read the bill?
* crossed:VBD - Have you read the bill?
Dice
* read:VB - Have you read the bill?
* crossed:VBD - Have you read the bill?
Min Edit Distance
* read:VB - Have you read the bill?
* crossed:VBD - Have you read the bill?
Local Alignment
* read:VB - Have you read the bill?
* crossed:VBD - The easiest way to influence your elected official is- you guessed it- writing a check.

Post - t3_1ee2m0
Cluster 1
The government was looking at tea parties groups tax-excempt status.
Were leftiest groups targeted as well?
Conservitive groups were targeted unfairly
Queries - bush, targeted
Cosine
* targeted:VBN - Yeah, they were right to target the 75 conservative groups out of the 300 total targeted groups.
* bush:NNP - What is being missed in this is not whether it 's conservative or liberal groups, the pertinent issue is the fact that the irs is being used as a threat to groups whom may have a legitimate protest against the current power structure.
Jaccard
* targeted:VBN - All of the 300 groups were conservative.
* bush:NNP - All of the 300 groups were conservative.
Dice
* targeted:VBN - All of the 300 groups were conservative.
* bush:NNP - All of the 300 groups were conservative.
Min Edit Distance
* targeted:VBN - All of the 300 groups were conservative.
* bush:NNP - It was n't 17/91 tea party groups.
Local Alignment
* targeted:VBN - `` the post you 're commenting on does n't claim that the irs did n't target *any* groups; it claims that the irs appears to have targeted liberal groups too.
* bush:NNP - What is being missed in this is not whether it 's conservative or liberal groups, the pertinent issue is the fact that the irs is being used as a threat to groups whom may have a legitimate protest against the current power structure.
Cluser 2
The IRS was targeting tea party and other community groups.
The tea party was breaking the law trying to apply for 501c4's.
The IRS admitted guilt!
Queries - groups, status
Cosine
* groups:NNS - You know more about what is irs policy than the president and the irs directors themselves.
* status:NN - You know more about what is irs policy than the president and the irs directors themselves.
Jaccard
* groups:NNS - I think it was the irs stating they did target tea party groups.
* status:NN - The upside is that this happened at the irs and not the white house.
Dice
* groups:NNS - I think it was the irs stating they did target tea party groups.
* status:NN - The upside is that this happened at the irs and not the white house.
Min Edit Distance
* groups:NNS - The irs was doing their job.
* status:NN - That is the premise of the scandal.
Local Alignment
* groups:NNS - Every country in the world has an agency like this, why the fuck would n't it profile groups?
* status:NN - The upside is that this happened at the irs and not the white house.
Cluster 3
Profiling occured with the IRS looking into tea party groups.
Many presidents have used the IRS to target they the opposition
Liberals are capable of making mistakes.
Queries - people, groups
Cosine
* groups:NNS - The irs issue was last on the list.
* people:NNS - The irs issue was last on the list.
Jaccard

* groups:NNS - The irs issue was last on the list.
* people:NNS - This was a few people at the irs, hardly a conspiracy.
Dice
* groups:NNS - The irs issue was last on the list.
* people:NNS - This was a few people at the irs, hardly a conspiracy.
Min Edit Distance
* groups:NNS - The irs issue was last on the list.
* people:NNS - This was n't some nixonian-style political abuse.
Local Alignment
* groups:NNS - It 's almost as if irs is keeping a closer eye on groups tied to the ideology not in power.
* people:NNS - The irs was doing their jobs for crying out loud.. this is all a bunch of bs.
Cluster 4
Liberals are ok with IRS targeting as long as it doesn't happen to them.
Obama should not be the only source for blame.
Extra scrutiny was placed on specific groups.
Queries - people, obama
Cosine
* obama:NNP - The irs is charged with checking the groups out.
* people:NNS - It could just be that the conservative groups stated goal is to destroy the epa.
Jaccard
* obama:NNP - The irs is charged with checking the groups out.
* people:NNS - The irs is charged with checking the groups out.
Dice
* obama:NNP - The irs is charged with checking the groups out.
* people:NNS - The irs is charged with checking the groups out.
Min Edit Distance
* obama:NNP - The irs is charged with checking the groups out.
* people:NNS - That is so not the same thing.
Local Alignment
* obama:NNP - The irs is charged with checking the groups out.
* people:NNS - It could just be that the conservative groups stated goal is to destroy the epa.
Cluster 5
The IRS started profiling Tea Party groups only because they found many dubious claims made by them
Conservatives are calling liberals out for not being open to their own shortcomings
Democrats and Obama are caving to Republican pressure.
Queries - groups, group
Cosine
* groups:NNS - The statement `` specifically singled out groups with tea party sounding keywords '' is very different to `` specifically singled out groups with tea party in the group name ''.
* group:NN - Irs targeting conservative groups is illegal.
Jaccard
* groups:NNS - The issue is they targeted conservative groups specifically.
* group:NN - Anyone who is going to start a group with that name is going to be right-leaning.
Dice
* groups:NNS - The issue is they targeted conservative groups specifically.
* group:NN - Anyone who is going to start a group with that name is going to be right-leaning.
Min Edit Distance
* groups:NNS - Irs targeting conservative groups is illegal.
* group:NN - The issue is they targeted conservative groups specifically.
Local Alignment
* groups:NNS - If a group wants tax free status and wants to be political then they can file as a 527 tax free entity] (http : //www.opensecrets.org/527s/types.php) or how the months tea party groups had to wait is nothing compared to the [5 years emerge nevada had to jump through hoops before being ultimately denied](http : //www.nytimes.com/2011/07/21/business/advocacy-groups-denied-tax-exempt-status-are-named.html? scp=3 & amp; sq=advocacy % 20groups % 20irs & amp; st=cse & amp; _r=0) because you said & gt; '' this is a blatant lie that a few(and only a few) liberals have made up because its the only excuse they can think of.

* group:NN - When it is a group of people who constantly lie making accusations, no i am not going to rush to judgement.
Cluster 6
There shouldn't be profiling
People are very biased agaist certain news sources
The IRS should make the distinction between a group who is tax exempt and who is a political action committee.
Queries - groups, bloomberg
Cosine
* bloomberg:NNP - Bloomberg is pro-fascism
* groups:NNS - Why should n't a liberal group receive the same scrutiny?
Jaccard
* bloomberg:NNP - Bloomberg is pro-fascism
* groups:NNS - Why should n't a liberal group receive the same scrutiny?
Dice
* bloomberg:NNP - Bloomberg is pro-fascism
* groups:NNS - Why should n't a liberal group receive the same scrutiny?
Min Edit Distance
* bloomberg:NNP - Bloomberg is pro-fascism
* groups:NNS - Not just conservative groups, either.
Local Alignment
* bloomberg:NNP - Perhaps you think bloomberg is a better source.
* groups:NNS - So we have a shit ton of tea party groups appearing out of nowhere overnight and they 're not supposed to get extra scrutiny for it?

Appendix B

Full Sample Reddit Post

Below is a sample Reddit post taken from the */r/technology* subreddit. It is referred to several times in the design chapter, Chapter 3, to help explain the pipeline with a real example. A snapshot of the post was taken from: http://redd.it/1cgncb on May 10, 2013. At the time the post contained 156 comments. Major portions of the web page were stripped to increase readability.

Parallella, the $99 Linux supercomputer (zdnet.com)

376 submitted 24 days ago by popstarpoop
156 comments share

sorted by: **best**

[–] **LHoT10820** 8 points 24 days ago

So biggest thing that I noticed was that they're claiming 90 GFlops of performance with under 5 watts of power consumed... Isn't that something to be excited about if it is true?

permalink

[–] **bitchessuck** 3 points 23 days ago

Actually, GPUs have similar GFLOPS/W efficiency nowadays. And a lot more GFLOPS/EUR.

permalink parent

[–] **iCanHelpU2** 7 points 23 days ago

It is, but in this thread people are most interested in rubbing their Wikipedia degrees in everyone's faces; rather than actually actually having a meaningful discussion :)

permalink parent

[–] **LHoT10820** 4 points 23 days ago

That's what I thought. Also can we stop dickwagging about how good GPUs are? Even with GPGPU they still can't do the same tasks as CPUs.

permalink parent

[–] **vithos** 1 point 23 days ago

I think you missed the part where previous-gen **CPUs** are > 90 GFlops. My i5-2500K (4.5 GHz) benchmarks at 118 GFlops without stripping down the OS/closing any programs. Their "equivalent GHz" number is off by 13.1x compared to a quad core; 3.3x off if you compare to a single core CPU AND that's the figure for their 64 core version, not the 16 core $99 version that got funded.

That said, there's a reason you don't see GFlops as the single measure of CPU performance, and what they're working on *could* be interesting anyway for other reasons. Unfortunately the exaggerations/bait and switch of this article makes me think I should just ignore the project instead.

permalink parent

[–] **LHoT10820** 1 point 23 days ago

I think you missed the part where it does CPU tasks and uses less than 5 watts. My boyfriends i7 has 110GFlops, but consumes 135 watts.

permalink parent

[–] **lanctotsm** 1 point 20 days ago

It doesnt do CPU tasks, you have to create special programs to use it. OpenCL lets you write programs in C but they have to be compiled on run time and transferred over to the parallel processing unit.

permalink parent

[–] **TriumphantTorpedo** 63 points 24 days ago

No. 45Ghz of Cortex A9 =/= 45Ghz of Core or 45Ghz of Vishera. Also, the bus to get data in and out of that chip is hilariously tiny, 1.4 GB/s, PCI-E 3.0, for instance, is ~ 16GB/s, and a Tesla Card will saturate that pared with a good CPU. This thing is a joke. It's an educational tool for teaching undergrads how to program on a highly threaded machine. Which, isn't a bad thing. The bad thing is that it's RISC and not X86 which is where people with that skill set (parallel programming) are required.

I find it redic, that a bunch of people calling themselves engineers managed to get nearly 900k by taking a bunch of COTS parts, an FPGA, and bolting it all together, then having the audacity to call it a super computer.

It's 2013, I thought we were past the whole Snake Oil thing.

permalink

[–] **nextwiggin4** 31 points 24 days ago

I invested in the project and I had thought the original project description was extremely fair. The idea wasn't to create a supercomputer, it was to build an extremely affordable multithreading practice computer. Something open source that students could learn on. They wanted it to be affordable so schools could buy lots of them easily.

They went out of their way to make it double as an internet TV so that casual investors could buy one and get a usable product (it has ethernet and HDMI). That was such a big selling point for the creators, that when I originally read about it was just an Apple TV competitor.

They did throw the "45Ghz" number out there, but they immediately clarified that it's not a very useful benchmark, it was simply meant as a way to compare the Epiphany to other similar chips out there. They knew the tech media would love that type of hyperbole and it would help the project.

This article is astoundingly unfair to what the original project was pitched as. The creators are hardly snake oil

salesmen. They raised $900k trying to build a teaching tool, and although it's not the only solution, it's the one that got people excited, and it's hardly a bad approach.

Also, it's not a terrible internet TV.

permalink **parent**

[–] **BuzzBadpants** 2 points 23 days ago

I'm afraid Nvidia might have beaten them to it on this one... Massively parallel courses based on CUDA are all the rage now, and you just need an nvidia graphics card in your computer.

permalink **parent**

[–] **eclectro** 2 points 23 days ago

> I'm afraid Nvidia might have beaten them to it on this one

Not really. The Parallella board consumes *magnitudes* less power (400 W PC versus 5 Watts) and costs a *magnitudes* less than a PC supporting an NVIDIA card ($800 versus $100).

As they scale up the number of processors it will eventually out compete NVIDIA GPUs.

permalink **parent**

[–] **lanctotsm** 1 point 20 days ago

The processors in the parallela pale in comparison to the flexibility of the processor architecture on the NVIDIA GPGPUS. While this processor has 32k of local memory for each processor NVIDIA can have 10 mb per processor, dynamically allocated, with over 1000 registers per thread and thousands of kilobytes of shared memory and local cache for ram accesses.

permalink **parent**

[–] **netraven5000** 32 points 24 days ago

Off-the-shelf parts like a 64-core Epiphany processor?

Also - I think you missed the point. They're trying to encourage a paradigm shift. Yes, we use multiple cores, multiple threads - but we don't really build things in a way that really takes full advantage of Amdahl's Law. That's what they're after - they want people to start designing programs to take better advantage of the fact that there are multiple cores.

> No. 45Ghz of Cortex A9

But it's their Epiphany chip that's doing the impressive stuff, not the A9.

> The bad thing is that it's RISC and not X86 which is where people with that skill set (parallel programming) are required.

Look at smartphones. They'd benefit from the reduced power consumption, and very few smartphones use an x86 instruction set.

Hell, look at game consoles and other entertainment devices. They'd see a pretty big benefit from this, and they typically *don't* use an x86 instruction set.

permalink **parent**

[–] **xtnd** 4 points 24 days ago

> but we don't really build things in a way that really takes full advantage of Amdahl's Law

Wait, you do understand that Amdahl's Law is a law **against** parallel computing, not in favor of it? It says that the performance increase when throwing cores at tasks decreases with every core you add.

And here's the thing about the world: Its not parallel. We can't just throw these boards in student's hands and suddenly be able to compute Fibonacci sequences in parallel. The *vast* majority of tasks are, to a large extent, fundamentally not parallelizable and never will be.

And moreover; most of the ones that are will never benefit from more than 8-16 cores. As in, the number of cores we have in our desktop computers. Look at Amdahl's Law. Even if 75% of the task is in parallel, we've nearly hit the core bottleneck with 16. And Adapteva wants *hundreds* on our boards? What good will that do?

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

> Wait, you do understand that Amdahl's Law is a law against parallel computing, not in favor of it? It says that the performance increase when throwing cores at tasks decreases with every core you add.

This really isn't my area of expertise, maybe I'm just talking out of my league - but Amdahl's Law is an argument for focusing your energy on certain parts of the code.

> And here's the thing about the world: Its not parallel. We can't just throw these boards in student's hands and suddenly be able to compute Fibonacci sequences in parallel. The vast majority of tasks are, to a large extent, fundamentally not parallelizable and never will be.

What do the numbers say about a computer that is splitting up tasks among its cores rather than simply using its cores to complete single tasks?

I'm not sure about your OS, but my OS supports multitasking. And uses it quite a lot.

permalink **parent**

[–] **narcoblix** 3 points 23 days ago

I'm not sure you quite understand how parallelization works. The examplt that xtnd gave was a good one: you can't compute fibonacci sequences in parallel.

A fibonacci sequence is one where to figure out what the next number is, you have to know the number that came before in that sequence. So to solve anything, you have to have a bunch of information that you calculated beforehand; you can't just start in the middle.

Because of that, you can't parallelize computing the fibonacci sequence.

That's just one example. There are a lot of things that are literally **NOT POSSIBLE** to make faster by using parallel processing.

permalink **parent**

[–] **netraven5000** 3 points 23 days ago

I understand how it works just fine.

And I did not refute his example. Instead I pointed out that typically our computers are switching between some number of operations. His example is a single thread that runs on one processor because it depends on the former two calculations. Okay. But what if I want to run Firefox while I'm crunching those numbers? Or what if I decide to listen to some music? Do I have to wait for those computations? No, I don't. These things can be done out of order and thus can be done in parallel with computing the Fibonacci sequence.

Let me rephrase: you're obsessing over the point that many individual algorithms/processes cannot be optimized, yet you're not accounting for the fact that our computers typically aren't performing just one task. The operation of the computer as a whole is something that can be done in parallel.

permalink **parent**

[–] **narcoblix** 2 points 23 days ago

It's true. There are gains that come from parallelization. I can listen to music, game, and read reddit (like right now ;p) because computers multitask (using a variety of methods including true parallel processing as well as asynchronous, and others).

However, the performance gains *in general* that come from parallelization are at best situational. Sure you may have a stuff going on in your computer at the same time, but 4-8 TRUE threads is about the max of where that pays off. So making the jump to *hundreds* of cores is not going to increase the performance by hundreds (or even tens in many cases).

So as general purpose devices, extremely parallel computers aren't highly practical. They do have awesome uses, and these are great teaching tools, but they'll not replace your desktop anytime.

On a slight tangent, there's a reason that this is causing such a reaction in the community. Because computers aren't just getting straight up faster at the same rate they were, lots of people are looking for the next magic bullet of computing.

Because that's what's sitting at the back of people's minds, everyone who is trying to gain any kind of support has to appeal to that; they are saying, or giving the sense that, they have the answer, the next magic bullet. That's the necessary aura that they have to put on for them to get media coverage, even if that's not what they really want to do. That's this particular case: this board is a great teaching tool, and has some excellent, if limited, practical uses.

However, it is not the magic bullet. That's what a lot of this media posturing and quarreling is: some people saying "Wow, maybe this is our savior!" while a bunch of other people come rain on that parade saying "No, this is not our savior. It's cool, but chill."

Sorry bout that, it's an tangentially related phenomena that I see a lot, and I just wanted to write it down I guess.

permalink **parent**

[–] **turnipX** 1 point 23 days ago

What about zipping? If we had 64 cores to zip/unzip files instantly couldnt we make much better use of hard drive space? Or perhaps rendering individual objects for physics calculations?

permalink **parent**

[–] **netraven5000** 1 point 23 days ago

It's worth noting - they have essentially two CPUs. I would imagine the ARM will be used in instances where it's really necessary that some *particular* process run faster than the cores in the Epiphany chip.

permalink **parent**

[–] **TriumphantTorpedo** 0 points 24 days ago

In order.

The Epiphany is * not * off the shelf, your right, I concede that. It is, however, not going to, nor is it part of, said "paradigm shift", unless that shift is towards "cell like" processors with tons and tons of gimped little cores.

Next, smartphones, things have pretty well settled out at quadcores with GPU acceleration, and those are even being badly utilized. This is because of a number of things, not the least of which is lazy coders. The expectation that you'll get these people, who can't even be bothered to write code for a quad core, to write it for a 16, or 64 core chip? That doesn't use the same A9 or A15 they are used to working with? Right.

Furthermore.

X86, as far as game consoles goes, is the future. The PS4 is getting a (very probably) Piledriver based AMD 8-Core, the new Xbox is rumored to have similar hardware, both X86 based, both with GP GPU acceleration capabilities, and both more in line (ie, identical) with how a desktop will act. Also, Intel has recently re targeted their Atom processors, the entire line is now x86 processors targeted at the RISC units in phones and "entertainment devices" (?), and it's benching as good as the current high end RISC quad cores. Here's the fun part, it's a single core with hyperthreading running at 1Ghz. Not too bad for a first attempt. (Look up the Acer Liquid C1, it's the first available so far)

Back to my original point.

If RISC was the future. If RISC HAD a future, this gizmo may make sense.

Since it doesn't. RISCs days are numbered. Intel is coming. and they are coming hard. They want this part of the industry, as does AMD, and I'm sorry, but once these two giants get into a performance/watt/$$$ war the first casualty is going to be the RISC chips. They'll be relegated to where they belong, in my 10 year old cousin's shitty solo phone that can text and make calls, and doesn't have a touch screen.

This silly contraption is too little, too late, and is going to be useful to a handful of people for a year or two. You aren't ever going to see one of these epiphany things in a real device, ever, and you'll never see more than a 5 core in a smartphone/tablet. Hell, they are just now starting to get a decent number of GOOD Tegra games out, and the Tegra has been around for like 36 months. (those 8 cores Google is promising in the N10 are not usable all at once. 4 are low powered, and 4 are high powered, it switches between the two dynamically, with only 4 running at any point in time.)

permalink **parent**

[–] **omnilynx** 13 points 24 days ago

> It is, however, not going to, nor is it part of, said "paradigm shift", unless that shift is towards "cell like" processors with tons and tons of gimped little cores.

Isn't that the whole point? That they're trying to encourage a massively multicore paradigm, since individual cores are starting to come up against physical limitations?

permalink **parent**

[–] **netraven5000** 7 points 24 days ago

> unless that shift is towards "cell like" processors with tons and tons of gimped little cores.

It's not about having little crappy cores. The shift is toward more cores in general. 64 cores is what this has.

What's happening is that these CPU manufacturers are finding that once they go beyond 2-3GHz, they're having trouble keeping power consumption and heat down. So that's why we're seeing them start to go with more *cores* instead.

> things have pretty well settled out at quadcores with GPU acceleration, and those are even being badly utilized

Isn't that kinda what I just said?

Yes. There isn't so much benefit. Because we need to use a different paradigm

> The expectation that you'll get these people, who can't even be bothered to write code for a quad core, to write it for a 16, or 64 core chip?

With quad-core, it's a lot of work before you see any benefit at all. Most of them reason that it isn't worth their time.

Also - this comment really shows your misunderstanding of the subject. No, you don't write code *for* the 64-core chip, and that's their point (well... I guess more accurately their point is that people *do* and they *shouldn't* and it's not necessary). You write it in a way that scales according to how many cores are available.

> X86, as far as game consoles goes, is the future.

I can't help but notice - those game consoles aren't so far in the future... and they're multicore... Huh, I wonder why they're multicore. It's almost like they know multicore is the way of the future...

I'd bet that they chose x86 because it's *familiar* rather than because the technology is superior.

> If RISC was the future. If RISC HAD a future, this gizmo may make sense.

Do you understand what the letters in the acronym "ARM" stand for?

Also - your argument makes no sense. You're saying this as though people will be writing all their code in assembly language that is specific to this processor. Which is probably not the case.

> you'll never see more than a 5 core in a smartphone/tablet.

I bet you will see it within the next 5 years. In fact, I bet you'll see 8 core tablets within the next 5 years.

> Hell, they are just now starting to get a decent number of GOOD Tegra games out, and the Tegra has been around for like 36 months.

Right, because this whole multicore thing is new, especially when it comes to phones. Not to mention, the idea of doing any kind of decently-intense gaming from a phone is fairly new.

Also - notice that you're making the first mention of mobile gaming.

permalink **parent**

[–] **the_leander** 3 points 24 days ago

> The international variant of the Samsung S4 uses an 8 core CPU - its divided into 4 A15 cores and 4 A7 cores. Huawei are also said to be taking this route with their octocore CPUs.

permalink **parent**

[–] **eclectro** 2 points 23 days ago

> X86, as far as game consoles goes, is the future.

Not necessarily. Intel no longer implements an x86 instruction set directly, but rather it gets translated into a different RISC microcode to implement multithreading inside the Intel chips.

It gets back to those lazy programmers you mentioned.

permalink **parent**

[–] [deleted] 24 days ago

[deleted]

permalink **parent**

[–] **netraven5000** 3 points 24 days ago

> Heavily multi-threaded programs are not rare, because there is no hardware to test on.

Correct, they are rare because with only four cores there is minimal benefit.

> They are rare, because most programming tasks are extremely hard/impossible to parallelize.

Most of these tasks are not what our computers spend their time doing. Also, *part* of the reason they're difficult is that they're working in a system that wasn't designed for high degrees of parallelism.

Also - you're dancing around the point. Making fire is not hard to us, but it was for the Neanderthals. Making muskets may not be hard, but the Native Americans didn't understand how to do it. Going to the Moon was hard, but we did it and now. This is hard because our current paradigms involve everything happening in a particular order and we tend to design things that way. But evidence suggests that continuing under our current paradigm will incur much larger costs in terms of how much power is required to achieve similar speed. Which is why we are *starting* to move toward multicore systems. Yes, it is hard. But people have been studying the potential for over half a century, and the conclusion is that we *must* do it eventually.

permalink **parent**

[–] **elementalist467** 1 point 24 days ago

The question is what application would drive adoption of highly parallel computing. Sony tried this approach in the PS3 with so much success that they have switched to x86 in the next iteration of the product. Regular user application code tends to benefit from high single process performance. For a solution like this to take off it requires a killer application. Something that it does categorically better that is relevant to users.

permalink **parent**

[–] **netraven5000** 3 points 24 days ago*

> Sony tried this approach

The Cell processor has one core that is based on PowerPC, and six cores that are not suitable for general-purpose computing.

> Regular user application code tends to benefit from high single process performance

Most code will not benefit directly from being split across multiple processors, true. But it will benefit greatly from having a relatively small number of processes running on the same CPU as itself.

> For a solution like this to take off it requires a killer application. Something that it does categorically better that is relevant to users.

Greater output with lower power consumption will be it. It may not be a big deal for desktops, but it will be for tablets and phones.

EDIT: Take a look at the type of figures they're talking about. The speed may not be that impressive - but the *power consumption* is way, *way* lower than what you'd expect from a conventional CPU. 2 Watts as a maximum? 0.1 Watts as a minimum? You're not gonna get that from your Intel chip.

It's on their Kickstarter toward the bottom. There's also a list of areas that would really benefit from massively parallel processing.

permalink **parent**

[–] **yanik** -1 points 24 days ago

> . for example my machine is running 1833 threads this moment

FYI, Intel CPU support 2 threads per core.

permalink **parent**

[–] [deleted] 24 days ago*

[deleted]

permalink **parent**

[–] **veive** 1 point 24 days ago

That's the rub isn't it? I don't know about you, but I don't just use one application at a time. The OS has several dozen threads running, the browser has a few, the IM has at least one, if I'm in skype/ventrilo/teamspeak/raidcall with someone that's another and so on. Will any one app that I use tap out 64 cores on it's own? no. Could all of the apps that I run happily tap out a core each and wind up using 64 of them? yep.

permalink **parent**

[–] **9542** 2 points 24 days ago*

FYI, he wasn't talking about hardware threads, he was talking about software threads, of which there is no limit in Windows 7.

Just because your CPU only has 2 "threads" per core doesn't mean that a process can't be using 1000 virtual threads.

Did you even bother to look at what he was talking about? "(just look up yours in task manager->performance)" ??? My machine is currently running 1030 threads.

And his point is that if you want to practice parallel programming, you don't have to have a parallel processor to do that, you can make as many threads as you want in a single core, you just won't gain performance boosts. The programming is the same, though, which is how you would get practice.

permalink **parent**

[–] **SteelChicken** 4 points 24 days ago

> .It's 2013, I thought we were past the whole Snake Oil thing.

But, but buzzwords! Linux! Supercomputer! Its like that very successful Raspberry Pi thing!

permalink **parent**

[–] **lanctotsm** 1 point 20 days ago

Did you see how many Tera Gigs it had?

permalink **parent**

[–] **WhiteZero** 2 points 24 days ago

> The bad thing is that it's RISC and not X86

I had always heard RISC is good...

permalink **parent**

[–] **turnipX** 1 point 24 days ago

Would it be possible given a different set of hardware to get 45 or so cores running well in parallel? Given a proper bus and everything, could it act as a stepping stone providing hardware to said device as well?

permalink **parent**

[–] **xtnd** 1 point 24 days ago

There have been multiple people across several subreddits that have tried to point this out, but no one believes them. Most of the people on reddit have absolutely no clue when it comes to these kinds of things, but they've managed to convince themselves that they're armchair experts on the subject.

permalink **parent**

[–] **IAmRoot** 1 point 24 days ago

Memory bandwidth is the most important thing for most applications these days. It doesn't matter how much compute you have if the vast majority of cycles are spent fetching data.

permalink **parent**

[–] **ihtkwot** 1 point 23 days ago

So, I don't want?

permalink **parent**

[–] **lanctotsm** 1 point 20 days ago

The parallela is not a 64 core A9 computer. It has a A9 processor and a 64 core proprietary architecture.

permalink **parent**

[–] **frickfrock99** 31 points 24 days ago

That "Equivalent to a 45Ghz CPU" analogy is ridiculous. Not even close to being true, how they'd come up with that?

permalink

[–] **ZankerH** 21 points 24 days ago

I'm guessing (number of CPU cores) * (CPU core clock). Silly and entirely non-indicative of actual performance.

permalink **parent**

[–] **netraven5000** 16 points 24 days ago*

Why guess when it says right in the article... it's based on the thing being capable of ~90 billion floating point

operations per second, versus an approximation of what it would take to create a conventional CPU that gets the same performance.

EDIT: OK, so I was wrong and you are correct. But they acknowledge in their Kickstarter page that there really isn't a good way for them to compare performance.

permalink **parent**

[–] **netraven5000** 9 points 24 days ago*

Are you serious?

It says right in the article how they got that. It delivers an estimated 90 GFLOPS of performance, and that's about equivalent to what a 45GHz machine would do (in theory).

Yes, it *is* true, *if* you understand what they're doing and what they're arguing. No, they are not saying this will be what you get from day one. What they are saying is that when you design your programs to take maximum advantage of all the cores, this is the kind of performance you can expect.

EDIT: I guess I was wrong. Anyway, there's really no good way to compare performance and the Kickstarter page addresses this.

> We have received a lot of negative feedback regarding this number so we want to explain the meaning and motivation. A single number can never characterize the performance of an architecture. The only thing that really matters is how many seconds and how many joules YOUR application consumes on a specific platform.
>
> Still, we think multiplying the core frequency(700MHz) times the number of cores (64) is as good a metric as any. As a comparison point, the theoretical peak GFLOPS number often quoted for GPUs is really only reachable if you have an application with significant data parallelism and limited branching. Other numbers used in the past by processors include: peak GFLOPS, MIPS, Dhrystone scores, CoreMark scores, SPEC scores, Linpack scores, etc. Taken by themselves, datasheet specs mean very little. We have published all of our data and manuals and we hope it's clear what our architecture can do. If not, let us know how we can convince you.

permalink **parent**

[–] **ZombieWomble** 11 points 24 days ago

You are, unfortunately, giving these people too much credit. They literally just multiply the CPU clock speed by the number of cores to get the 45 GHz number. From their kickstarter page:

> Once completed, the 64-core version of the Parallella computer would deliver over 90 GFLOPS of performance and would have the the horse power comparable to a theoretical 45 GHz CPU [64 CPU cores * 700MHz]

For a more realistic comparison in terms of calculation speed, it should be observed that modern high-end CPUs already match or better the quoted 90 GFLOPS performance - i7s have been reporting > 100 GFLOPs for some time, at clock speeds far below 45 GHz.

permalink **parent**

[–] **netraven5000** 2 points 24 days ago

Either way - what's impressive isn't the speed so much as the power consumption. 0.1W-2W? You'd be lucky to get below 60W with a regular Intel processor.

permalink **parent**

[–] **ourocks3** -1 points 24 days ago

Lol pulling 120 Gflops on a 3570k will fetch ya atleast 120-150 watts. Check out r/gamingpc they have charts and whatnot showimg performance.

permalink **parent**

[–] [deleted] 24 days ago

[deleted]

permalink **parent**

[–] **eras** 6 points 24 days ago

Because they have luxuries like being able to run different code on each core and have features like conditional gotos and loops without crashing the performance.

permalink **parent**

[–] **Paran0idAndr0id** 2 points 24 days ago

And, those GPU's aren't ARM processors, they are power hogs, and they are expensive.

To compare power, the 8800gs gets 3.77 GFLOPS/W, this chip gets 90GFLOPS/5W = 18 GFLOPS/W, a 5-fold increase in efficiency (assuming both are performing perfectly as reported, which know will rarely be the case).

permalink **parent**

[–] **eclectro** 2 points 23 days ago

The power thing will become *huge* as Moore's law bumps against the wall i.e. the increase in computing power (pun not intended) will include how much it costs to flip a bit, rather than just the number of transistors that are implemented.

permalink **parent**

[–] **thomas41546** 1 point 23 days ago

Lets use a more recent example: GeForce GTX 680M gets about 19 GLOPS/W.

permalink **parent**

[–] **Paran0idAndr0id** 1 point 23 days ago

That's at least $300-500, doesn't have an attached ARM/x86/x64 processor to run an OS, and doesn't come bundled with motherboard and RAM (Well, technically you generally can only buy the M lines attached to laptop motherboards, so you could argue that it *does* come with 'bundled' etc etc, but you get the point).

permalink **parent**

[–] **thomas41546** 1 point 23 days ago

Well instead of paying ~$100 for 90GFLOPS 64 cores you are getting 1900GFOPS (100W) for ~$500. Seems to me a much better deal in performance per price -- even if you factor the cost of an arm board in, say $100 in price.

permalink **parent**

[–] **Paran0idAndr0id** 1 point 23 days ago

That's not a $500 card though, that's a $500 laptop upgrade. You need the other 1500 in laptop to be eligible to even buy it.

permalink **parent**

[–] **ramate** 2 points 24 days ago

Yep, I had 2 GTS 250s (basically a shamelessly overclocked 8800) which could easily pull 130-140 GFlops apiece.

permalink **parent**

[–] **netraven5000** 4 points 24 days ago*

Because you can't use your GeForce 8800 GS as a CPU.

EDIT: Just want to add - your GeForce 8800 GS runs at 105W at the max. Compare that to Parallella's chip which runs at 2W at the max.

EDIT 2: Their Kickstarter explains why they call it a supercomputer.

> The Parallella project is not a board, it's intended to be a long term computing project and community dedicated to advancing parallel computing. The current $99 board aren't considered supercomputers by 2012 standards, but a cluster of 10 Parallella boards would have been considered a supercomputer 10 years ago. Our goal is to put a bona-fida supercomputer in the hands of everyone as soon as possible but the first Parallella board is just the first step. Once we have a strong community in place, work will being on PCIe boards containing multiple 1024-core chips with 2048 GFLOPS of double precision performance per chip. At that point, there should be no question that the Parallella would qualify as a true supercomputing platform.

permalink **parent**

[–] **austeregrim** 1 point 24 days ago

Maybe you can't.

permalink **parent**

[–] **bitchessuck** 1 point 23 days ago

You can't use an Epiphany core as a generic CPU either. These are specialized cores with certain weaknesses that need custom programming to get good performance out of. Just like GPUs.

permalink **parent**

[–] **netraven5000** 1 point 23 days ago

> You can't use an Epiphany core as a generic CPU either.

Each core of the Epiphany chip is a generic RISC CPU with its own memory, yes.

> These are specialized cores with certain weaknesses that need custom programming to get good performance out of.

Depends on what you mean. If by "custom programming" you mean "your application must be specially designed in order to run faster" - yes, that's true. If you mean "it won't work unless you rewrite your program" - no, that's not true.

permalink **parent**

[–] **nextwiggin4** 2 points 24 days ago

In the description for the project they immediately clarified that it was a terrible form of a benchmark and didn't really mean anything. They explained that they just didn't really have a very good way of giving an accurate benchmark and wanted to convey it's performance somehow.

permalink **parent**

[–] **asdfgasdfg312** 3 points 24 days ago*

Its based on multiple cores, so that if you multithread efficient enough you could run a program *like* it's being runned

on a 1core 45Ghz. It's not like you can run 1 instance of skyrim at lightspeed and more like you can run 45 instances of skyrim at the same time without you losing any computer power.

permalink **parent**

[–] **ixid** 5 points 24 days ago

What would happen if I loosed my computer's power?

permalink **parent**

[–] **Jelal** 14 points 24 days ago

Then you would have to make it tighter?

permalink **parent**

[–] **mobile4ever** 4 points 24 days ago

Badda-bing!

permalink **parent**

[–] **electricalnoise** 3 points 24 days ago

Just hope you already runned your backup program so you don't loose anything.

permalink **parent**

[–] **ixid** 2 points 24 days ago

Running made my computer looser in the first place.

permalink **parent**

[–] **Natanael_L** 1 point 24 days ago

Did it drop it's case?

permalink **parent**

[–] **ixid** 2 points 24 days ago

Only it is lower case.

permalink **parent**

[–] **asdfgasdfg312** 1 point 24 days ago*

well it's not really losing power as much as its forcing your cpu to do useless read/write pulses etc. Don't really know how to translate it in to correct computer science terms in English.

permalink **parent**

[–] **IHateEveryone3** 2 points 24 days ago

The word you were looking for was *lose*, not *loose*. Hence the questions are not authentic, they are laughing at you.

permalink **parent**

[+] ***asdfgasdfg312*** *comment score below threshold* (3 children)

[–] **hudders** 10 points 24 days ago

Supercomputer in the traditional sense of the word. Before this thread becomes choked with people thinking something else.

permalink

[–] **netraven5000** 6 points 24 days ago

From their Kickstarter:

> Why do you call the Parallella a supercomputer?
>
> The Parallella project is not a board, it's intended to be a long term computing project and community dedicated to advancing parallel computing. The current $99 board aren't considered supercomputers by 2012 standards, but a cluster of 10 Parallella boards would have been considered a supercomputer 10 years ago. Our goal is to put a bona-fida supercomputer in the hands of everyone as soon as possible but the first Parallella board is just the first step. Once we have a strong community in place, work will being on PCIe boards containing multiple 1024-core chips with 2048 GFLOPS of double precision performance per chip. At that point, there should be no question that the Parallella would qualify as a true supercomputing platform.

permalink **parent**

[–] **lecrazedutch** 3 points 24 days ago

Can I play games on this?

permalink

[–] **hobomouthwashparty** 1 point 24 days ago

More importantly how well will it run said games and can it function well as a minecraft server?

permalink **parent**

[–] **ferret_guy** 4 points 24 days ago

It can run a minecraft server but it is not designed for this purpose, unless you wrote your own client the minecraft server would not take advantage of the Epiphany Multicore Accelerator, defeating the point.

permalink **parent**

[–] **ummwut** 2 points 24 days ago

AS far as I can tell, since the minecraft server software is Java, then rewriting the JVM to use the extra cores when threading would be the only change needed.

After that's done, then you can go over the server code for multicore optimization, but I'm sure (I hope) Mojang has already done that.

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

I believe most JVMs already do that, it's just a question of whether or not Minecraft is multithreaded.

permalink **parent**

[–] **ummwut** 0 points 23 days ago

We should probably send an email to whoever is dealing with minecraft now, or hell, even a tweet to jeb.

permalink **parent**

[–] **netraven5000** 0 points 23 days ago

I don't know who that would be and honestly I don't care much about Minecraft.

permalink **parent**

[–] **iCanHelpU2** 1 point 23 days ago

Not as far as I know. Having that many core is taking the current architecture and throwing it out the window. I'm sure special software is needed to take full advantage of the system's capabilities.

That being said, many modern GPU's can outperform this, especially the newest cards, that are able to work in the Teraflops. The advantage here is simply the power to price ratio :)

permalink **parent**

[–] **TheHy-Mag** 3 points 24 days ago

Scientific computing is the area most in need of parallel processing, and this piece of hardware is not likely to be adopted. Enthusiasts might set up test clusters, and find them wanting. We already use large Infiniband-connected clusters and highly optimized code with MPI protocols. For a lot of algorithms scalability is the primary concern.

OpenMP is typically the second layer in a hybrid parallelization, and works on shared memory. It doesn't scale beyond the number of cores within an individual processing node.

permalink

[–] **IAmRoot** 3 points 24 days ago

Also, the most common bottleneck these days is moving data around. This thing has terrible memory bandwidth. Things like collective communications are very expensive. For example, we are hitting the limit of scalability on FFT-based algorithms, which is going to suck. Debugging a program run across several million cores is a nightmare as well.

permalink **parent**

[–] **knylok** 4 points 24 days ago

I want this, but I don't know what I would do with it.

permalink

[–] **dogismyname3** 4 points 24 days ago

Doorstop.

permalink **parent**

[–] **aperrien** 3 points 24 days ago

I could use it in my work for data mining and analysis.

permalink **parent**

[–] **cass1o** 1 point 24 days ago

How would GPgpu computing compare?

permalink **parent**

[–] **aperrien** 1 point 24 days ago*

Some of the things done with Oberon and Haskell have been incredibly useful to me. Having an upcoming alternative wouldn't hurt, as some of the contortions to get that code running can be very difficult.

Edit: Spelling

permalink **parent**

[–] **do_you_hate_me** 1 point 24 days ago

Wanna go into more detail?

permalink **parent**

[–] **netraven5000** 2 points 24 days ago

It's basically a Raspberry Pi except it's a lot more powerful. Think of it that way.

permalink **parent**

[–] **do_you_hate_me** 0 points 24 days ago

> It's basically a computer except it's a lot more powerful. Think of it that way.

permalink **parent**

[–] **netraven5000** 7 points 24 days ago

A regular PC? No, this isn't much more powerful than a regular PC.

permalink **parent**

[–] **do_you_hate_me** -2 points 24 days ago

Really? Then why is it called a supercomputer?

permalink **parent**

[–] **netraven5000** 3 points 24 days ago

Why do you call the Parallella a supercomputer?

> The Parallella project is not a board, it's intended to be a long term computing project and community dedicated to advancing parallel computing. The current $99 board aren't considered supercomputers by 2012 standards, but a cluster of 10 Parallella boards would have been considered a supercomputer 10 years ago. Our goal is to put a bona-fida supercomputer in the hands of everyone ***as soon as possible but the first Parallella board is just the first step***. Once we have a strong community in place, work will being on PCIe boards containing multiple 1024-core chips with 2048 GFLOPS of double precision performance per chip. At that point, there should be no question that the Parallella would qualify as a true supercomputing platform.

permalink **parent**

[–] **Tennouheika** 1 point 24 days ago

Bitcoins

permalink **parent**

[–] **knylok** 2 points 24 days ago

That was the only thing I could think of.

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

Unless you want to buy an ASIC, you're way too late to get into the Bitcoin game. No matter what you buy (unless it's an ASIC).

permalink **parent**

[–] **Tennouheika** -1 points 24 days ago

Well honestly I think bitcoins are either a scam or just a terrible investment.

permalink **parent**

[–] **aaa801** 2 points 24 days ago

How can you think bitcoin is a scam?

permalink **parent**

[–] **netraven5000** 1 point 23 days ago

I think there's room to argue it might be a pyramid scheme. The people who got in early might make a lot of money - people who get in now might never recover their investment and may lose all of it.

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

Maybe both. One way or the other I'd say at this point it's definitely ill-advised to invest any real money in them. You probably won't gain much, and the potential loss is pretty huge.

permalink **parent**

[–] **aaa801** 0 points 24 days ago

Litecoin

permalink **parent**

[–] **imahotdoglol** 2 points 24 days ago

ahh litecoin, where the amount being sold is 100x the amount people are trying to buy.

permalink **parent**

[–] **Brodie123** 3 points 24 days ago

Is this any good at mining bit coins?

permalink

[–] **goatfucker9000** 15 points 24 days ago

If it ever becomes commercially available it will be massively outperformed by the ASIC miners that will be hitting the market shortly

permalink **parent**

[–] **raven12456** 2 points 24 days ago

I'm tempted to pre-order a Butterfly labs ASIC miner, but it's so far away who knows if it'll be worth it.

permalink **parent**

[–] **Lentil-Soup** 2 points 24 days ago

Supposedly, current pre-orders will be shipping by the end of July. Who know, though...

permalink **parent**

[–] **raven12456** 1 point 24 days ago

Even if they're trading around $50/BTC, one of the 5Ghash units would pay itself off in about three weeks. As long as there aren't any major changes in price or difficulty.

permalink **parent**

[–] **Lentil-Soup** 1 point 24 days ago

I know. I really want to buy a Jalepeno, I'm just afraid I won't get it until July 2015.

permalink **parent**

[–] **raven12456** 4 points 24 days ago

Makes me wonder if all of these miners are done, but they're sitting in the company offices mining away...and getting "delayed".

permalink **parent**

[–] **Isakill** 1 point 24 days ago

I've already pre-ordered the 5Gh one, and this is exactly what i'm worried about.

288 bucks is quite an investment to wait so long for.

permalink **parent**

[–] **cass1o** 1 point 24 days ago

Wouldn't everyone getting an asic cause the difficulty to skyrocket?

permalink **parent**

[–] **raven12456** 1 point 24 days ago

I think it will. Why I added the disclaimer at the end ;) . Especially since there is a long line of preorders before us, who knows what will happen the first few months that these come out.

permalink **parent**

[–] **ShouldBeZZZ** 1 point 24 days ago

1) Even if they do ship out, it's too late for you to get much profit or any profit at all 2) Butterfly labs is sketchy as fuck, I wouldn't do business with them if I had a million dollars to waste.

permalink **parent**

[–] **RobNine** 3 points 24 days ago

Buying those never made sense. I mean cost of unit + electric bill + time to mine 1 Bitcoin =/= Value of a Bitcoin

permalink **parent**

[–] **sweartobatman** 8 points 24 days ago

Some of us don't pay electricity ;-)

Also I'm guessing it's a matter of investment. Some people believe the value of Bitcoins will just keep rising...

permalink **parent**

[–] **RobNine** 3 points 24 days ago

I would like to see a solar powered one.

permalink **parent**

[–] **goatfucker9000** 3 points 24 days ago

The megahash/$ ratio will probably be comparable our better than these "super computers" though.

permalink **parent**

[–] **RobNine** 1 point 24 days ago

Most definitely.

permalink **parent**

[–] **xtnd** 1 point 24 days ago

That's not true at all. As an example, the unreleased Bitforce Jalapeno can hit over 4.5 GH/s. Currently, mining can get you ~$0.50 / 24 Hours @ 100 MH/s. That means the Jalapeno could generate about $20 per day at current market rates. It uses 4.5W of electricity, which is less than a lightbulb. And it costs $150; that'll be paid for in less-than a week.

Once it comes out, the profitability will decrease. But even if it drops by a full order of magnitude, it'll still be more than enough to remain solvent.

permalink **parent**

[–] **batquux** 1 point 24 days ago

Might be good for litecoin though.

permalink **parent**

[–] [deleted] 24 days ago

[deleted]

permalink **parent**

[–] **lllIIIIIIIIIIIIIIIII** 6 points 24 days ago

Man, I just looked up a GTX690 and it does 5621.76 GFLOPS, compared to the 90GFOPS this CPU does. That's nuts.

permalink **parent**

[–] **lecrazedutch** 1 point 24 days ago

Would the producers of these ASIC machines ever sell them if it was more profitable running them themselves?

permalink **parent**

[–] **Natanael_L** 2 points 24 days ago

Sell one at a high markup, build two more for that money. Repeat.

permalink **parent**

[–] **buge** 1 point 24 days ago

If they get pre-orders then they don't have to put in their own capitol, they can have pure profits with no risk.

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

In terms of how fast it is, it's not any better than modern GPUs.

In terms of how much power it consumes, yes.

permalink **parent**

[–] **Inscothen** 1 point 24 days ago*

unless there was an update, the $99 parallella kickstarter board only has 16 Epiphany cores and 2 ARM cores.

The real performance will come when they scale past 64 cores.

permalink

[–] **netraven5000** 2 points 24 days ago

They have two different versions. The $99 one is 16 cores and the $199 one is 64 cores.

permalink **parent**

[–] **Inscothen** 1 point 24 days ago

Yes, but when they talk about the $99 version in the article, they don't mention it's only the 16-core version. It's built on an older manufacturing process(65nm) and has way less performance per watt. The $200 64-core didn't happen for the kickstarter because they didn't reach their stretch funding goal.

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

I think they still plan to do a 64-core version, though. Probably after people get the 16-core version and start really making use of it.

permalink **parent**

[–] **based2** 1 point 24 days ago

https://news.ycombinator.com/item?id=5557985

permalink

[–] **TomSwirly** 1 point 24 days ago

Gosh, I could use this as a compile farm - *except* for the too-small memory.

64 cores, 1 gig? That's about 16MB of memory per core - not so much...

permalink

[–] **netraven5000** 1 point 24 days ago

They said it's a limitation of the ARM processor they're using as a host, and future versions will use a better processor that can support more RAM.

permalink **parent**

[–] **RonMexico69** 1 point 24 days ago

so is this a working computer for 100? or just the processor or motherboard or what? I'm confused.

permalink

[–] **mavensift** 1 point 24 days ago

Anyone playing with one yet? Is this better then the Raspberry Pi?

permalink

[–] **georgeo** 1 point 23 days ago

Everybody here knows that a couple hundred bucks on a video card gets you *tera*flop performance that blows this away.

permalink

[–] **[deleted]** 1 point 23 days ago

Is this just a raspberry pie?

permalink

[–] **nafe19** 1 point 23 days ago

thats cool .. only 99 !!!

permalink

[–] **avisetia** 1 point 23 days ago

Another XBMC machine.

permalink

[–] **NaChoBizness** 1 point 24 days ago

This would have been a totally cool idea... over a decade ago.

For the same money, you can buy an off the shelf GPU which probably outperforms this system significantly in throughput, and has more mature programming tools.

Alas, who knows... this may be an interesting approach from a hobbyist point of view. But I doubt this is going anywhere.

permalink

[–] **netraven5000** 2 points 24 days ago

A GPU cannot power your entire OS, though.

Their intent is to be basically a massively parallel Raspberry Pi - much faster than Raspberry Pi, and rather than teaching kids about a single-core CPU, it teaches kids about a 16 or 64-core CPU.

Whether or not this project will take things there remains to be seen of course, but this probably *is* where computing is headed. They're really not kidding when they say it takes a lot less power to achieve approximately the same performance.

permalink **parent**

[–] **lanctotsm** 1 point 23 days ago

The biggest issue I have is that they are using a paralyzed architecture for their cores. No common memory, each has their own segment, terrible bandwidth, proprietary instruction set, and buzzword buzzword buzzword. The modern Cuda core or Stream processor in a GPU has so many technologies that can be taken advantage of by very powerful programming tools, and the cards themselves can be very very cheap. The next generation of the tegra chip will even be CUDA capable. No software out on the market will be able to use this epiphany processor I guarantee it.

permalink **parent**

[–] **netraven5000** 2 points 23 days ago

Uh, it said right on the page it'll work with OpenCL and other technologies.

permalink **parent**

[–] **lanctotsm** 1 point 23 days ago

Work and work well are two very different things.

permalink **parent**

[–] **netraven5000** 1 point 23 days ago

Work well is something we won't know until someone gets one and uses it.

permalink **parent**

[–] **ScroteHair** 1 point 23 days ago

Actually I'm pretty sure if you tried you could use a GPU to power your OS.

permalink **parent**

[–] **batquux** 0 points 24 days ago*

Is this going to be like the "$25" raspberry pi was really $50? Well, $60, if you actually buy it.

permalink

[–] **goatfucker9000** 3 points 24 days ago

There are two versions, the first to be released was the $35 model B that included onboard ethernet and a second USB port. I ordered one from Newark/Element 14 and it came with free shipping, so I paid exactly $35. It was on back-order at the time, so i had to wait about 3 weeks for it, but I got it for the advertised price. The same applies to the currently available model A which goes for $25. Although if you don't have a spare phone charger and an SD card lying around they will be extra.

If you want a $25 computer and have an extra micro USB charger and SD card lying around then you can actually have a working Raspberry Pi for $25.

permalink **parent**

[–] **vilette** 1 point 24 days ago*

a link please

permalink **parent**

[–] **goatfucker9000** 1 point 23 days ago

http://www.newark.com/jsp/bespoke/bespoke7.jsp?bespokepage=newark/en_US/landing/raspberry-pi/rasp-pi-accessories.jsp&CMP=KNC-G-SLOC

permalink **parent**

[–] **vilette** 1 point 23 days ago*

thank you so much for this link, on farnell, which is European newark, they sell it 36,23$ on wich you add 21% tax so 43$. and it's always out of stock,except if you order with a box (total 55$) ??

I'd like to live in murica, twice as much for the same money

permalink **parent**

[–] **netraven5000** 1 point 24 days ago

Except this is a lot more powerful than Raspberry Pi.

permalink **parent**

[–] **strategosInfinitum** 0 points 24 days ago

Just take my money!

permalink

[–] **7nkedocye** 0 points 24 days ago

Too bad a graphics card with more GFLOPS cost like $40.

permalink

π