PROJECT SUMMARY

Overview, Key Words, Subtopic

The combination of increased processing power and large amounts of digitized text data opens new possibilities for word prediction and machine understanding of natural language. This project involves building a real-time autocompletion service for general English text. The system will organize and analyze large amounts of data sourced from English text and use it to predict upcoming words based upon a snippet of text. Fact extraction and other NLP methods will be used to understand the deeper meaning of phrases and provide autocomplete suggestions beyond what is available from current systems. This project is in the fields of natural language processing and computational linguistics, and involves big data and potentially machine learning. The relevant subtopic area is IT1: Artificial Intelligence, Machine Learning, and Natural Language Processing.

Intellectual Merit

This Small Business Innovation Research Phase I project will help advance knowledge of how to make computer systems effectively understand and respond to human language through improving such systems ability to anticipate upcoming words and to match meaning to sentences. Given a sequence of words, current systems struggle to provide useful predictions of upcoming words to users, or to match words to concepts and deeper meanings. This project will start by using n-grams of varying sizes from the Google Books Ngram dataset to build prediction models based on algorithms developed. Next, other sources of n-gram data will be investigated, with custom datasets built if appropriate. Additionally, fact extraction and other methods may be used to glean insight from the actual meaning of words. Because of the complexity inherent in these problems, speed is a major usability issue that this project seeks to improve. Providing real-time results (that is, results where the delay is imperceptible or effectively zero to the user) is a particular goal. This will require innovation in the methods of storage and access of the data used to ensure acceptable response times, as well as careful selection and refinement of the prediction algorithms.

Broader Impact

Better computer understanding of natural language will allow humans and computers to work together more efficiently and more effectively in a wide variety of situations. Fast autocompletion will increase the speed in which people can communicate to computers their intended meaning. Voice recognition capabilities can be improved if systems know what words are likely to be spoken after other words. Fast autocompletion furthermore allows systems to pre-load results based on predicted complete queries or sentences, thereby lowering average response times. It is normal today for people to be constantly interacting with computers, and fast, effective autocomplete will improve these interactions and have a positive impact for many people many times a day.

ELEVATOR PITCH

Computers have led to incredible increases in productivity in a wide variety of ways. The increasingly realized ability of computers to understand natural language offers the potential for additional increases in productivity. Autocompletion of natural language is useful as both a user interface element for a user to directly interact with and as a backend component to improve the speed and accuracy of services. Our project involves building autocomplete that is faster, more accurate, and more flexible. The incorporation of our autocomplete into applications and services will result in improvements in the efficiency and effectiveness in which people use them. Unlike current autocomplete systems, ours will provide better predictions by using the deeper meaning of phrases and understanding the user's intent more completely.

There are a wide variety of potential applications for our technology. Autocomplete can be used for search, as millions of people see each day when using a major search engine. Specialized searches, or searches focused on certain types of queries, such as questions, can also benefit from autocomplete. Writing applications can use autocomplete to help users write quicker and find inspiration. Autocomplete can also be used in service backends to improve recognition of natural language. For example, if a speech recognition program is having trouble understanding some of the words spoken by a user, it can use autocomplete and the words it does recognize to make an educated guess and improve accuracy. The users of our autocomplete and products and services it is integrated into could be anyone who interacts with computers, which is almost everyone in today's world.

We are using the state-of-the-art Google Books Ngram Corpus to create our autocomplete. The English portion of the dataset contains ngrams from over 4.5 million books and over 460 billion words. Using this expansive dataset ensures the algorithms and techniques we develop work well with common sources of text. For future work we will consider the use of other data sources, including potentially custom sources for specific specialized purposes or proprietary sources from clients. The algorithms and techniques we develop will not rely on the use of a specific dataset to be effective.

Simple autocomplete can be built just by looking at frequencies of phrases, but we will also be using fact extraction, synonym matching, and related techniques to understand the deeper meaning of phrases, which will allow us to provide better predictions in ways current systems cannot do. This technology will allow our system to be effective when many edge cases and language quirks occur, even if they would cause issues in more basic autocomplete systems.

Speed is also a focus of our autocomplete. Results must appear to be real-time for autocomplete to be useful in many cases. Our system will use an innovative combination of storage techniques and processing algorithms to ensure predictions appear sufficiently quickly, even while providing better predictions by understanding the deeper meaning of phrases and the user's intent.

The Commercial Opportunity

There are several potential areas where our project presents commercial opportunity. One major opportunity is assisting companies and other organizations that have digital services or applications that require interacting with people using text or speech. Autocomplete, in particular, is useful to companies offering a wide variety of services and applications. Companies offering voice recognition services, or applications that include voice recognition, can improve their accuracy with autocomplete. Autocomplete can be used where companies offer search or question answering features in their products. Autocomplete can also be used in writing applications to suggest upcoming word and phrases to the end user. For companies with these types of products and more, our system could improve performance and revenue potential. Because of the large amounts of research, effort, and expertise needed to make high quality autocomplete, it would be more efficient for companies to work with us than to develop their own solution. Companies using our technology would save themselves the time and expense of learning how to create and optimize complex high quality autocomplete systems, ensuring they get the best results possible and maximize the positive experiences of their own customers.

Companies of different sizes and needs are looking for different types of help with autocomplete, so we would provide both a autocomplete system that could be licensed and easily integrated into a client's own systems by their own programmers, and arrangements where our experts work directly with a company to build custom autocomplete solutions into their products.

There is also significant opportunity to develop our own products with autocomplete as a central feature. To help people with writing, we can develop plugins and extensions for word processing applications that provide autocomplete, or even a specialized writing application. We could also create keyboard applications for smartphones with features aimed at improving typing accuracy and speed on touchscreens. Programs to search, index, and help people sort through their own documents would also be a good application of our natural language processing expertise.

Our autocomplete system uses enhancement techniques beyond what is used in current autocomplete systems, and these techniques may make our system useful even in areas that current autocomplete systems do not perform well in. Our enhancement techniques allow our system to understand and provide effective predictions for inputs that are strangely worded or formatted, which could open up commercial opportunities with clients who have not found other autocomplete systems to be effective.

There are a number of risks that could potentially harm our commercial viability. It is possible we will be unable to get our autocomplete methods to be as effective as we want without compromising our goals of real time speed, reasonable storage sizes, and high quality results. Our enhancement techniques may not be able to improve results as substantially as we had hoped for some use cases, which could make our system less attractive compared to autocomplete built using only historical data. We may need to make compromises to ensure that our system meets the goals that are most important, but these compromises could make our product less appealing. Another risk may lie in potential clients not understanding the potential benefits of autocomplete (or better autocomplete) in use in their products, or in potential clients prioritizing other aspects of their products.

Our revenue potential is somewhat unclear. Autocomplete and natural language processing can be used in so many different digital products and systems. Our operations could include integrating autocomplete into systems in new ways, improving the performance of autocomplete and NLP systems currently in place, and making new products. Our revenue potential thus depends on our ability to convince other companies to buy our product or hire us to work with them, and our company's ability to meet that demand.

Societal Impact

Our autocomplete system will be a part of other, larger systems and products, and so much of the societal impact of our product depends on what specific products it is integrated into. Our product makes other services and products more effective and more powerful, and so could increase whatever societal impacts those products have.

Good autocomplete can be very helpful for those with accessibility issues, and our product could make technology easier to use for people affected by a variety of accessibility issues. Those who have trouble using a keyboard quickly (perhaps because of decreased dexterity or ability to control hand movements) will benefit from autocomplete saving keystrokes, and could also benefit from speech recognition backed by autocomplete. People who struggle with spelling, whether because of a condition such as dyslexia or any other reasons, can benefit from autocorrect systems that can use our autocomplete system.

Our autocomplete system will not be reliant on any one dataset of historical data, so we will be able to at least partly accommodate some languages other than English. It will be a challenge to ensure that we have the proper datasets for other languages, particularly obscure languages and languages with relatively small fluent populations, and to make sure our techniques work well with other languages. Languages with large sets of characters, such as Chinese and Korean, will not be likely to work under our original system, but we may be able to make changes to support them. Some changes may also be needed to take full advantage of diacritical marks in languages where they are common.

Autocomplete is an important part of moving human-computer interaction toward greater use of natural language and quicker understanding of a user's intention. The societal impact of humans working with computers more effectively is hard to overstate considering the prevalence of computers in modern life and continued advances in computing technology. By increasing the effectiveness of human-computer interactions, our autocomplete system will increase the utility of computers in many areas of life and help drive further advances to increase the effectiveness of which we can use computers.

Technical Discussion

Making an autocomplete system is not too hard. Making a useful autocomplete system - one that is sufficiently fast, uses acceptable amounts of disk space and memory, and returns useful predictions - is hard. In addition to the normal challenges of autocomplete systems, ours will be improving result quality using advanced natural language processing and other techniques, and so has the additional challenge of meeting the same usability standards as current autocomplete systems while performing additional processing.

There are a number of steps involved in turning raw ngram data (as formatted in the Google Books dataset) into data ready to be used for autocompletion. The first step is to clean the data. We remove ngrams with tokens that are not wanted (commonly punctuation and other symbols that are not words) and remove unwanted annotations from words (such as part of speech annotations). We also make everything lowercase. Due to the size of the raw ngram datasets, it is necessary to divide the raw files into chunks for processing. Within each chunk, each line is parsed to separate the different fields. The ngram tokens are cleaned, and if the ngram is to be kept, desired fields are saved to a temporary list. At the end of each chunk, the temporary list is sorted alphabetically and saved to a temporary file.

Next, the temporary files are combined and sorted to make one large sorted file. Even if the ngram data originally comes sorted (as the Google Books ngram data does), after cleaning the data sorting is likely to become necessary again.

We then evaluate the ngrams to try and form additional predictions through a variety of methods. The specific methods can vary depending on the dataset being used and the type of predictions desired. These enhancement techniques allow our system to understand input that may be strangely worded, oddly formatted, or otherwise ill-suited to work smoothly with traditional autocomplete methods.

Categorical classification is one possible enhancement method, where ngrams are categorized and predictions are improved based on shared upcoming words or other characteristics of all ngrams classified together. Synonym substitution may be used to create additional ngrams based upon synonyms. Supplementary datasets can be used in a variety of ways to produce additional predictive ngrams or to help determine which ngrams should be suggested to the end user.

To end the preprocessing stage, predictions for words and phrases are exported as a gzipped JSON file (to allow for easy importing while also saving storage space). Decisions are made as to which word and phrases to export (for example, by limiting the number of predictions for each word to a certain number).

Then, the JSON file is opened, and all or some of the data within is imported into a MongoDB database. Within the database, we are currently creating a MongoDB collection for each two letter sequence, and putting the data for all words that start with those two letters into that collection as a MongoDB object formatted similarly to the JSON exported from the processing step.

One challenge of our project is to provide fast results despite the additional processing required to return predictions beyond simple historical data. We will be trying to do some of our additional processing during the data processing phase (so it only needs to be done once) instead of each time a new completion request is made, but may be unable to do this for all of the techniques we use. (Additionally, doing this poorly could lead to a major increase in disk space requirements and a corresponding increase in latency, which we must carefully manage.)

Another challenge is to ensure the techniques we use to improve our autocompletions actually provide useful improvements in realistic scenarios and settings. We must ensure we focus on those techniques which provide either at least small improvements in lots of use cases for our system or large improvements in a small set of important use cases.

Our project is innovative because it goes beyond current autocomplete standards, which focus solely on parroting back exactly what has followed a word or phrase in past examples, to using creative methods to provide a more expansive, useful, and accurate set of completion suggestions. Historical data of which words follow which words is undeniably useful, and we will use it as the core of our system. But we believe we can go further than that and, using our enhancement techniques, deliver useful autocompletions that go beyond what is possible from simple historical data.

We also hope the results of our analysis of input to our system will help applications and services provide additional functionality. While our methods to understand the user's intention at a more abstract level will help improve our autocomplete results, we also will return relevant insights to applications so they can use them for other purposes.

Our first objective was to build basic versions of each large part of the system. We built a basic program to perform the preprocessing on data, a program to import the data into MongoDB in the right format, and a program to query the MongoDB database. We implemented data cleaning and sorting in the preprocessing step, and used a small subset of data to test all steps and ensure they worked together properly.

The next objective, and an important technical milestone for us, is scaling to a full load of data and building real-time autocomplete based solely off historical data while also remaining in reasonable boundaries of disk space and memory use. This may be more challenging than for a normal autocomplete system due to the breadth and size of the Google Books ngram dataset we are using as our source. To reach this milestone, we will be gradually adding more and more data to our systems and testing response times and system performance.

Once we get this purely historically sourced autocomplete working, we can begin adding our more complex techniques to the system that will enable is to provide improved results. In addition to upcoming word predictions, we may output additional information that we extract, as that information may be of use to programs and services using our autocomplete.

Although our autocomplete system is meant to be used as a part of other systems, and so direct user interface problems are not really a concern for us, we will likely build a basic graphical user interface for testing and demonstration purposes. We may also make our autocomplete predictions available as an API to offer another option of access to services using our system.

Revisions

Based on feedback from classmates and my own review of my work, I made a number of changes. I restructured the writing to clearly delineate different parts of the project, and to put each larger idea in its own paragraph. I reworked a number of awkwardly worded sentences, including replacing repetitive words with alternatives. I elaborated on the upcoming objectives we are trying to achieve.