Slide 1:	Hi everyone, my name is Steven Felix, and I'm excited to share with you search suggester. A tool I created to improve stack overflow queries.
Slide 2:	First, I'd like to tell you about Jamie. Jamie is a newly hired data scientist at a company that pays her good money to answer important business questions. Her time is valuable. But sometimes Jamie spends too long trying to find answers to even the simplest coding questions.
Slide 3:	For instance, how to create a column in a pandas dataframe. Jamie might scour the top search results for her answer, only to find middling relevance. Little did she know that even the slightest change to her query, for instance changing "create" to "add", would yield much different set of results that are more relevant to what she is looking for. Users like Jamie, particularly those who may be new to coding or to a particular language, do not always know the words or terminology that other people are using.
Slide 4:	To solve this problem I created Search Suggester (go to demo)
Demo	Search Suggester takes in as input a query you might submit to stack overflow and outputs 5 alternatives. What makes these alternatives useful is that they more closely reflect word choice and word combinations that are most likely to occur in the title of a Stack Overflow question, thus improving the likelihood of finding an answer to your question. *As a side note: You may notice that some of the words appear to be identical or overlapping. For instance "create" and "creating" are the same word, no? It turns out that stack overflow search engine is really REALLY not smart, and even small changes in verb tenses or inflections, noun pluralization alter the search results and their rankings. For this reason, search suggestor considers these distinct words.]* *[Go to alternative example] In cases of very rare words or very specific words (eg package names), sometimes search Suggester does not provide suggestions that are helpful to a user. In such cases, a Query Constructor that makes available a list of similar words for each original word in the query. The user can then click the terms they want to use in a new query in order to populate the search bar, and then they can click "to stack overflow" to see search results.]*
Overall approach and	So how does search suggester generate these improved queries?

algorithm explanation	It works in two steps: 1. Generating potential alternatives 2. Ranking them by their likelihood of occurring Both of these steps can be achieved through a natural language processing algorithm called *word2vec*. At a high level, what word2vec does when given a body of text is learn the *meanings* of words contained in the text [as well as the *conceptual relationships* between words]. It does this by examining the contexts in which each word occurs. Words that occur in similar contexts are inferred to have similar meanings. For instance "add" and "create" often are used interchangeably, so a word2vec model would infer similar meanings to these words. In learning the word meanings, word2vec also learns the relationships of words and the contexts in which they occur. This means you can calculate a probability of an entire phrase occurring. For instance, a properly trained model should find that this first phrase is more probable than this one, because '*pandas'* is more likely to co-occur with 'python' than with 'c+'. It's ability to learn this correctly, however, and to generate suggestions that are aligned with Stack overflow questions -- requires the proper training corpus - one that includes the terminology and word usage that a person searching Stack over Flow might use.
Data	Thus, what better training data than stack overflow titles themselves? I parsed over 17M question titles from a Stack Overflow data dump of all user content. As expected, these titles include the language that search suggester needs to know in order to make appropriate suggestions, including references to package names, function calls, files like robots.txt. *Optional: In order to properly model words and terminology as they actually exist in these titles, I performed a minimal amount of text preprocessing, though I did remove stop words, as these would not disrupt the most important terms in a title.*
Constructing suggestions	Once trained on this corpus, the word2vec model can be put to work **generating suggestions**. I do this by taking each individual word from an input query, and using the trained model to find the top 5 most similar words. Next, I construct full phrases by taking every possible combination of these individual words. For instance, this might be one combination. But not all of these constructed suggestions are made equal. Some combinations are more likely to match the questions/answers found on stack overflow.

	To find these, I use the word-context probabilities built into my trained model to calculate the probability of each phrase. With this in hand, I return to the user only the top 5 most probable phrases.
Validation	In order to evaluate different aspects of search suggestor's performance, first I constructed 100 test queries. I fed each of these test queries to Search Suggester and retrieved the top 5 suggestions. Comparing the probability of the test queries and suggested queries, I find that the suggestions are 29x more probable, under my trained model. This means that they more better reflect the word usage and word combinations of the Stack Overflow titles in my training data. Next, I submitted each of these suggestions, as well as the original query, to a true stack overflow search, and I scraped a number of relevant metrics from each results page, including titles, votes, and answers. Comparing the titles, I found that the results from my suggestions overlapped with results from the original queries by only 25%. This means that my suggestions are having a meaningful impact on search results, leading users to a mostly distinct set of quetisons and answers. Furthermore, looking at the distinct results, my suggestions tend to lead to more popular questions (20 more votes on average), meaning that they are considered more important by the stack overflow community. My suggestions also lead to questions with slightly fewer answers. But this may not always be a bad thing, as popular questions sometimes have one or two very good answers, at which point the community stops posting new answers. Overall, these results indicator that search suggester may be a useful tool to improve the stack overflow search efficacy, particularly for people new to coding or learning a new computing language.
About Me	Now, a little about me: Trained as experimental psychologist. Addressing questions about close relationships and well-being. For instance, why does perceived criticism from family members predict relapse for depression and schizophrenia? I'm also an avid baker and had the opportunity to work professionally for a bit. I love to run and train for road races. However, the majority of my time is actually spent wrangling and trolls and superheros.