Automated English Style Tutoring

Anish Tondwalkar
Juan Arrivillaga
Prateek Agrawal

Inspiration

- The English language have become the *de facto* language for communication.
- Non-native speakers face difficulty in expressing their thoughts in written English
- Access to human tutors is expensive, and often logistically difficult
- We present an automated approach to english tutoring

Contributions

- Incorrect or stylistically incongruent text highlighted with a heatmap
  - Redder text -> more likely to be incorrect

Contributions

- Incorrect or stylistically incongruent text highlighted with a heatmap
  - Redder text -> more likely to be incorrect
- We make sentence-level suggestions to replace "bad" text with better

Contributions

- Incorrect or stylistically incongruent text highlighted with a heatmap
  - Redder text -> more likely to be incorrect
- We make sentence-level suggestions to replace "bad" text with better

Usability studies have found that using heatmaps to visualize errors performs well in helping humans deal with suggestions that may have a negligible false positive rate.

Related Work

- Deep Learning Language Models
- Automated Tutoring

Deep Learning Language Models

- Deep Learning approaches in NLP involve Word2vec model by Mikolov et al.
- This model learns word embeddings in vector space.

Deep Learning Language Models

- Deep Learning approaches in NLP involve Word2vec model by Mikolov et al.
- This model learns word embeddings in vector space.

There are 2 basic implementations

- Skip-Gram Model
- Continuous bag of Words(CBOW)

Skip-Gram Model

Skip-Gram predicts the context words from target words

CBOW Model

CBOW predicts the target words from the context words

An Example

For example "The quick brown fox jumped over the lazy dog"

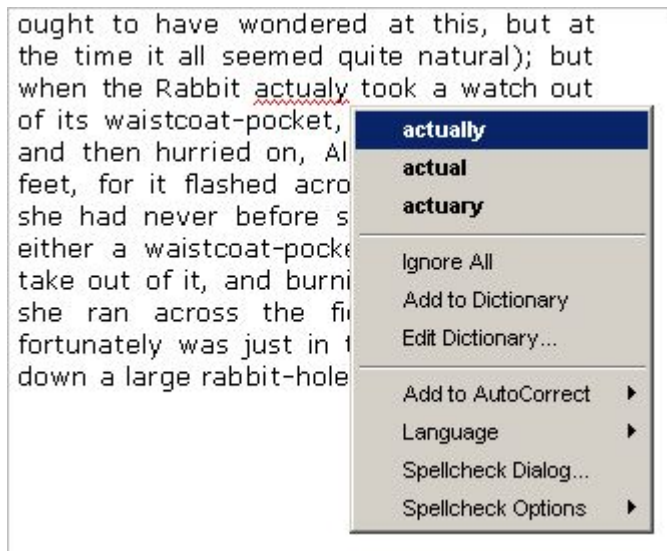Taking the context window as 1. So, we get (context, target) pairs as

([the, brown], quick), ([quick, fox], brown), ([brown, jumped], fox), …

For skip gram model we will predict context from target

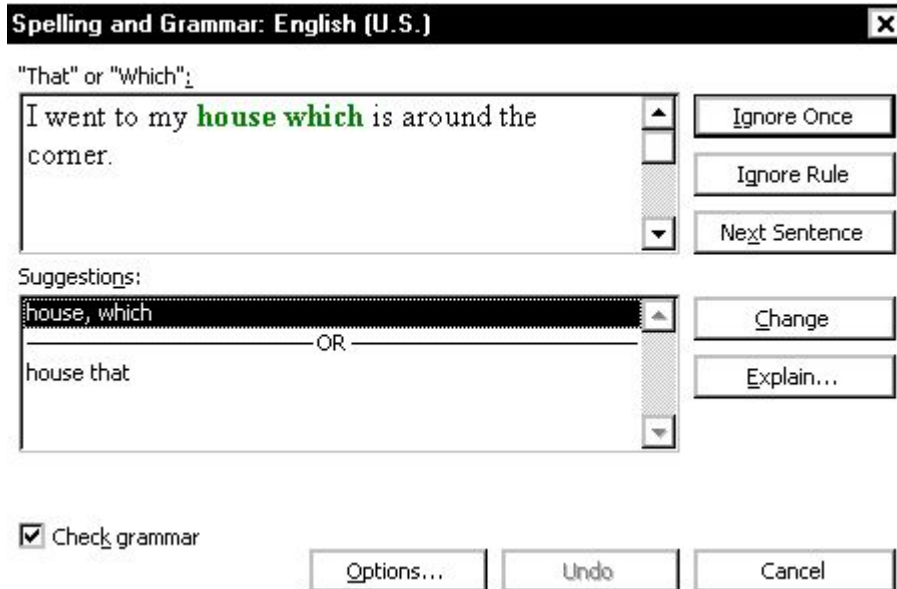(quick, the), (quick, brown), (brown, quick), (brown, fox), …

Automated Tutoring

- Spell checkers

Automated Tutoring

- Spell checkers
- Rule-based Grammar checkers (commercially successful, e.g Grammerly)

Automated Tutoring

-   Spell checkers
-   Rule-based Grammar checkers

No prior work using statistical techniques

Automated Tutoring

- Spell checkers
- Rule-based Grammar checkers

No prior work using statistical techniques.

Can we do better with statistical text mining techniques?
Can we convey writing style?

LIVE DEMO!

Data sets

- Gachon
- Reuters
- Wikipedia

Datasets

Gachon Dataset

http://koreanlearnercorpusblog.blogspot.kr/p/corpus.html

- Consist 2.5 million tokens
- 25000 participants answering 20 questions like:

1. What topics should people avoid during small talk? Why?
2. Have you ever done something that was really thrilling or frightening? What did you do? Etc….

Datasets

Wikipedia dataset

- 420 wikipedia articles were scraped in relation to the questions asked in gachon dataset to train the Word2vec model
- 34000+ sentences.

Datasets

Reuters Dataset

- No. of documents 7769
- No. of sentences 37000+
- Open source reuters articles developed and compiled by personals from reuters ltd.
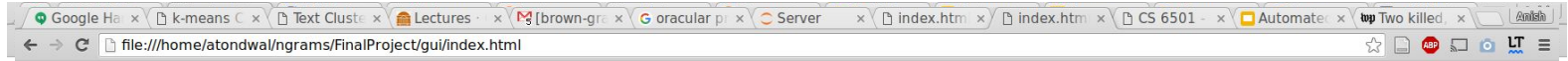
Contributions

Text Highlighting

Model of native English based on Reuters corpus and Wikipedia to compare native English to the non-native English found in the Gachon corpus.

Text Highlighting

- Log-likelihood of each ``chunk'' of text, given its context, to determine coloring.
- Experimented with several different granularities of chunking and settled on sentence-level chunking.
- We try a few different generative language models:
  - naive n-grams
  - skip-n-grams
  - CBOW

Screenshot

Text Suggestion

The text suggestion problem is a difficult one, but simple models can serve as a jumping-off point for the rest of our work, and to provide a standard of comparison.

Approach

- Generate sentences using N-gram Language model(we used 5 gram language model)
- Use Word2Vec Skip-Gram Model to calculate the vector representation for each newly generated sentence
- Use cosine similarity between old sentence and newly generated sentences (threshold=0.8)

Screenshot

The apocalypse began in a cubicle.

Its walls were gray, its desk was gray, its floor was that kind of grayish tile that is designed to look dirty so nobody notices Smith-Teller and I am twenty-two years old. I was fiddling with a rubber band and counting the minutes until my next br

"AR-ASH-KON-CHEL-NA-VAN-TSIR," I chanted.

That wasn't a hidden transcendent Name of God. That w

about two cents, and cost a small portion of my dignity.

Consider Instead:

districts had been deprived of any means of subsistence because

Ignore this type of error

Examples…

"AR-ASH-KON-CHEL-NA-VAN-TSIS," ordered my computer, and I complied. "AR-ASH-KON-CHEL-NA-VAN-TSIS," I said.

The little countdown clock on my desk said I had seven minutes, thirty nine seconds until my next break. That made a to Hebrew phrase "arei miklat" meaning "city of refuge" summed to 459. There were six cities of refuge in Biblical Israel, thr either side of lunch. None of this was a coincidence because nothing was ever a coincidence.

Exploring Distributed Vector Models

- Word2Vec creates word embeddings.
- Words that share the same contexts are semantically similar.
- For sentence similarity,
  - We sum the vector representation of all words in a sentence
  - By summing them we now would not be subjected to word order
  - Use cosine similarity between vector to calculate their similarity(threshold = 0.8)

Evaluation

- Cross-validation of text ratings for text highlighting with different models and different chunking lengths
- Didn't find a significant difference in generalized performance.
- No easy way to evaluate word2vec, as several factors like amount and quality of data used, size of vectors, training algorithm influence the quality of word vectors.

Future Work/Shortcomings

- Text suggestions are computationally expensive, we used random search!
- Try to find a more principled approach to searching for similar phrases.
- Will output some irrelevant sentences which have similar word2vec vectors
- Integrate suggestions with rule-based grammars
  - (We just suggest unpunctuated strings of words)

Finally - we need access to better "novice-English" corpora. They exist, but they are expensive. Free ones are usually very biased (only a few prompts).

Questions??

Extra Slides

Language Models

- Generative probabilistic models of a language that gives a probability that a string is a member of a language.

Language Models

- Generative probabilistic models of a language that gives a probability that a string is a member of a language.

Uses of Language Models

- Speech Recognition
- Generation
- Context sensitive recognition

Machine Translation

- IBM Models
  - Simple model

- Mikolov et al.'s Skip-Gram and Continuous Bag of Words (CBOW) models
  - Learns linear mapping between vector spaces of languages.
  - Excellent Performance
  - Available as the word2vec library!

Evaluation

Traditionally, correction tasks are manually evaluated, which can be arduous for the researcher. Recently, crowd sourcing platforms like Amazon's Mechanical Turk have made this approach more scalable \cite{park_automated_2011}. However, several~\cite{skoufaki_exploratory_2009,lee_building_2011,wang_english_2011} methodologies for the evaluation of English writing are available in the literature. These generally focus on the evaluation of natural language produced by non-native language speakers learning the language, but several approaches seem promising for the evaluation of scientific writing. Jiang et al.~\cite{jiang_automated_2015} build a extensive corpus of novice English writing by Chinese students learning English as a second language.

Vector Space Representation

- Represents words in a continuous vector space where semantically similar words are mapped to nearby points.
- The distance between the vectors tells the relationship of words/sentence