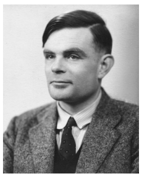
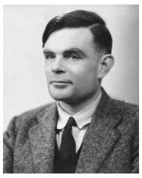
Sentence Generation using Two Models

Final Project in the course DD2380 at KTH

Group 61

K. Hannesson	J. Jóhannsson	E. Ahlsén	J. Andersson
Agust 20	January 12	BIRTHDATE3	February 10
hannesso@kth.se	jokull@kth.se	edvarda@kth.se	jonand8@kth.se



October 14, 2015

Abstract

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

NOTE

- The following sections are arranged in the order they would appear in a scientific paper. We think that these sections need to be there and written. However, these are only guidelines and if you think that some of these sections or subsections are irrelevant to you, please feel free to remove them. Similarly, if you want to include more sections or subsections please go ahead. Also feel free to rearrange them according to your convenience, but keeping some common sense (eg. Introduction cannot come after Conclusions).
- *Introduction, Related Works, Experimental Results, Discussions, Summary* are sections that MUST be contained.
- In the section of your *Method*: please do not list your project as log book entries, please talk about the final method you want to present to us. Talk about the method scientifically or technically and not as ”I did this...” ”Then I tried this...” ”this happened....” etc.
- Do not paste any code unless it is very relevant!
- The section *Contributions* is a place to express any difference in contributions. The default assumption is that you all agree that all of you had an equal part to play in the project.
- We suggest that you try to write this as scientifically as possible and not simply like a project report. Good Luck!
- Please remove **this** NOTE section in your final report.

1 Introduction (1–2 pages)

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

1.1 Contribution

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

1.2 Outline

Bla bla bla bla bla bla bla Section 2 bla bla bla bla bla bla bla bla bla Section 3 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla Section 4 bla bla bla bla bla bla Section 6 bla bla bla bla bla

2 Related work

In general, according to [**?**], NLG systems are constructed after performing requirements analysis whereby the goals of the NLG system are defined. Varied requirements have produced several different NLG systems over the years. Indeed, the project team designed the two NLG approaches after performing such a requirements analysis.

In [4] three methods for text generation are examined. One of the models examined makes use of n-gram models. The paper explores methods for making use of the observed usage of the language in a corpus, but without manually constructing a grammar. A generation of attribute templates was the novel approach by [4] where a sequence of placeholders is constructed. The placeholders are then filled in by the three proposed models. This idea inspired the inferred grammar approach and the structure of this project whereby two models are explored.

Several textbooks explore the topic of language modelling using n-grams and part of speech tags. Works such as [3], [5], and [2] provide excellent tutorials to language modelling and were used extensively in this project.

3 Our method

The group came up with the idea to compare two different approaches to generating text from a corpus, both including grammar but in different ways. To be able to compare them at the same level both approaches used the Brown corpus and trigrams, with fallback to bigrams allowed. A few smoothing techniques were picked for use in both approaches.

The first approach was to create a model that included both word and grammar information. This was achieved by generating trigrams where each word was a tuple of the word and its associated Part-Of-Speech (POS) tag. A trigram from "the man walked" would be (the, DET), (man, NOUN), (walked, VERB))

The second approach separated grammar and words into two models which were used in sequence to generate text. The grammar model provided what POS tag should be next, and the word model provided a word

for the given tags, both using trigrams. This would take "the man walked" and create the trigrams (DET, NOUN, VERB) for the grammar model, and (DET, NOUN, man) and (NOUN, VERB, walked) for the grammar-word model.

The smoothing techniques chosen were:

- Maximum Likelihood Estimate
- Laplace smoothing
- Expected Likelihood Estimate
- Simplified Good-Turing Frequency Estimation

3.1 Implementation

We split into pairs with each pair implementing their model. We decided to go with Python 3 for the implementation because we knew the NLTK package would provide us with the necessary building blocks to construct and test the two models. These included

- Brown corpus
- Treebank Part of Speech Tagger (Maximum entropy)
- Punkt Tokenizer Models
- Mappings to the Universal Part-Of-Speech Tagset
- Conditional Frequency Distribution and Conditional Probability Distribution classes
- classes for each of the above mentioned smoothing techniques

4 Experimental results

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

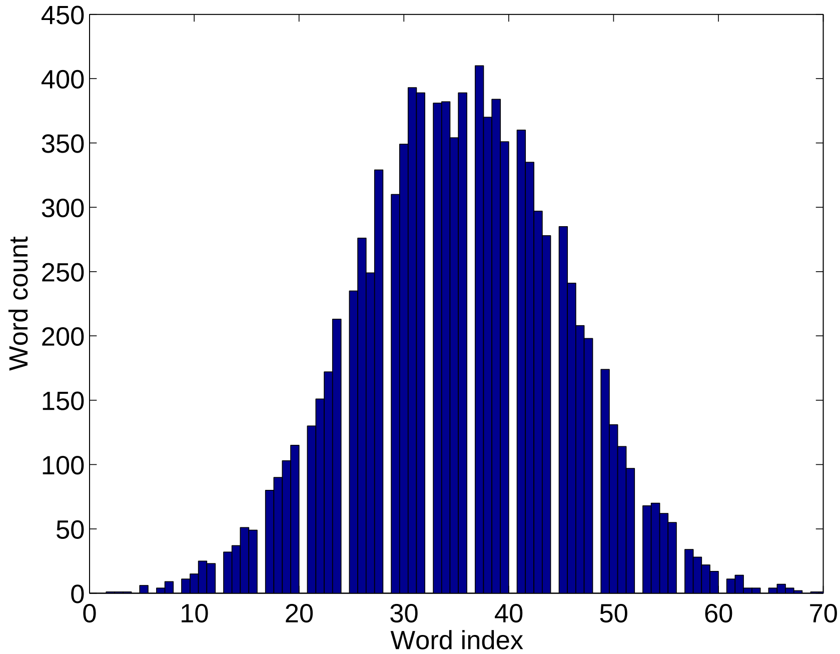


Figure 1: A description that makes browsing the paper easy and clearly describes what is in the picture. Make sure that the text in the figure is large enough to read and that the axes are labelled.

4.1 Experimental setup

Each model was used to generate 5 sentences per smoothing method for a total of 40 sentences. 5 sentences were also picked from the Brown corpu[1] s. All the sentences were mixed together randomly and then added to a survey which allowed participants to rate each sentence on a scale of 1 to 5 whether it was created by a human or a computer, with 1 representing definitely a human and 5 definitely a computer. Apart from the sentence scoring we only asked if the participants were native English speakers or not. The survey solution chosen was QuestionPro.

4.2 Experiment ...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Bla bla bla bla bla Figure 1 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Bla bla	Bla bla	Bla bla
42	42	42
42	42	42

Table 1: A description that makes browsing the paper easy and clearly describes what is in the table.

Bla bla bla bla bla Table 1 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

5 Future Work

6 Summary and Conclusions

7 Contributions

We the members of project group 61 unanimously declare that we have all equally contributed toward the completion of this

References

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

[2] Stanley F. Chen. An empirical study of smoothing techniques for language modeling. Technical report, 1998.

[3] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[4] Adwait Ratnaparkhi. Trainable methods for surface natural language generation, 2000.

[5] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems.* Cambridge University Press, New York, NY, USA, 2000.

[6] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach*. Number ISBN 978-0-13-207148-2. Pearson Education, 3rd edition, 2010.