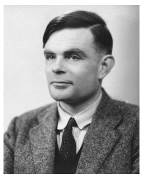
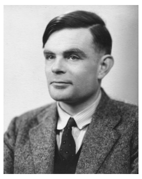
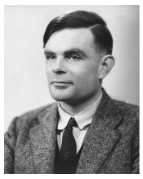
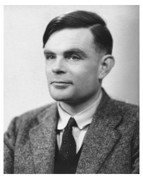
Carl Bildt Tweets: A comparison of regular and constrained Markov chain for text generation

Group Ain't intelligent

Viktor Björkholm	Jesper Bränn	Daniel Duberg	Jakob Tideström
92-11-17	92-09-30	93-01-13	90-10-04
viktorbj@kth.se	jbrann@kth.se	dduberg@kth.se	jakobti@kth.se
			

Abstract

Skriv sist, nr vi vet vad vi har skrivit om (y)

1 Introduction (1–2 pages)

This paper aims to develop an understanding on refining natural language text generation. Natural Language Generation (NLG) is an area of research within the field of Artificial Intelligence. The aim is to generate text that is semantically correct in order to make communication with computer systems more natural and understandable for users.

Within this paper we show the difference in quality of two different approaches to text generation. One of these approaches is using Markov chains or more commonly known as n-grams. These n-grams take n words in sequence and uses a corpus of text to guess what the most probable next word is. Using a larger n means that more text is copied straight from the corpus, however this also means that there is a higher likelihood that the text being generated is meaningful. We will contrast this method with using constrained Markov chains. The main constraint of the Markov chains is that two words following each other will have the same sequence of part-of-speech as the corpus. Part-of-speech is a concept within NLG that divides a text into the different linguistical categories of the words within it.

We aim to show that using this constraint upon the Markov chain, sentences will have a greater diversity but still be as semantically correct as just using a n-gram. Since the problem with larger n:s within n-grams is that text is copied straight from the corpus, the constraint will help us create sematically correct scentences without taking word sequences straight from the corpus.

To be able to show differences in these two approaches we generate twitter messages, so called tweets. We build upon the work by Barbieri et al., 2012 to implement our own constraints.

1.1 Contribution

A lot new research involves the constraining of Markov chains to produce text that fits different molds than just text generated from a corpus. It is important to be able to generate text that is

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 5 bla bla bla bla bla

2 Related work (1–3 pages)

Our work on the constrained Markov models are built upon Barbieri et al. [2012] work, where the authors generated lyrics from different artists using Markov models **of the second grade (a bi-gram)** with constraints. These were to be generated in a specific style and with a rhythm. However those criteria were not necessary for our work but we utilized the knowledge of constrained Markov models from them.

Saker vi behver svara p i detta stycke:
Vad vet vi om hur dem anvnde constraints i sitt arbete? Vi verkar ju inte veta ngot om det eftersom vi bara gissar oss fram :D
Har vi ngot mer att basera vrat arbete p? En vetenskaplig text r ju bra, men om vi lst fler s r ju det rtt bra.

3 My method (1–4 pages)

To show the method it is easier to consider the Markov chain to be of the first order, a so called unigram. This means in a limited corpus, for example:

"Rolf has a dog."	Noun → verb → singular quantifier → noun
"Rolf owns a dog."	Noun → verb → singular quantifier → noun
"Rolf can not walk his dog."	Noun → modal → adverb → verb → pronoun → noun

Would generate a chain that looks like the following, with the edges being the probabilities for the specific transitions:

In the figure the specific part-of-speech is included in the states. If we now add constraints from a transition matrix (built up from POS-analysis of the corpus) from the same corpus we are given this new chain:

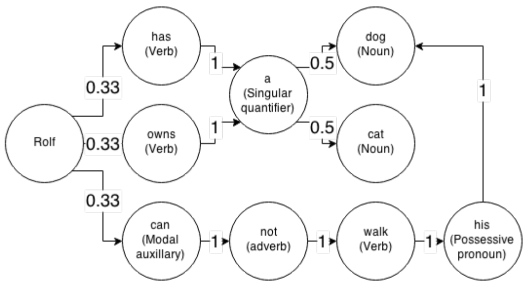


Figure 1: Unigram of the limited corpus

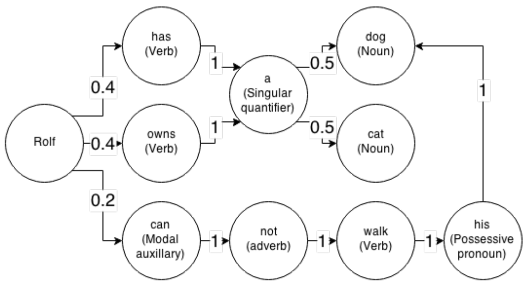


Figure 2: Unigram with constraints applied

We see here that since both ”has“ and ”owns“ are verbs they are more probable to occur than ”can“, this is reflected in the new edges. This happens because the part-of-speech ”verb“ is more likely to follow NNP according to our transition matrix, and thus ”has“ and ”owns“, who are both verbs become even more likely to follow NNP (Rolf).

In i a similiar way we will weigh in the constrants from the transition matrix in to a bi-gram, to weigh in the probability for a specific part-of-speech to follow as well as a specific word.

Corpus:

In order to generate tweets, we first discussed the problem regarding the semantical corectness of a generalized tweet. The average level is according to our own experience far from sematically correct, which isn't a problem in understadabillity for an experienced twitter user, but is a problem for our POS who would not recognize the words. Even if it would give it an ”unknown“-tag, we would not be able to predict any kind of results.

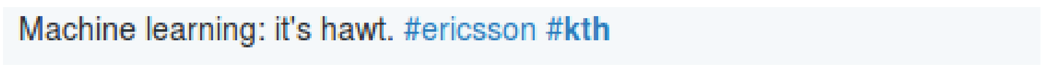

Machine learning: it's hawt. #ericsson #kth

Figure 3: An example tweet

The Markov chain is built up with the use of n-grams of the second order, so called bigrams. This means in a limited corpus like for example “Rolf has a dog.“ “Rolf owns a dog.“ “Rolf can not walk his dog.“ Would generate a bigram that would look like this:

Rolf	→ has
has	→ a
a	→ dog
Rolf	→ owns
owns	→ a
a	→ cat
Rolf	→ can
can	→ not
not	→ walk
walk	→ his
his	→ dog

We see here that the probability of nice coming after the words ”a“ and ”very“ is 100%, since that is the only combination that happens. However, the probability of ”town“ coming after ”very“ and ”nice“ is only 50%.

Our method consisted of building a transition matrix as a unigram of parts-of-speech (POS) together with a bigram with actual words from our corpus of text. From the transition matrix we generated a constrained transition matrix based on the amount of number of words we wanted the tweet to contain **(and other criteria)**. The constrained matrix was generated using the method described in the work by Barbieri et al. [2012]. We generalized the method and made it work for our transition matrix even though it consisted only of parts-of-speech.

The different constrains for creating a tweet that we had to consider were that they can not be longer than 140 characters, they have to end with an end-symbol (dot, exclamation mark or question mark) and they should probably have a reasonable minimum length. When iterating through a corpus we are using a POS-tagger to identify the different types of words and coding to be able to build a transition matrix for the sequences for the different word types (the probability for a noun to be followed by a word for an example). The next step is to implement our constraints on the transition matrix. The constraint for the length of the tweet caused us problems since we only know the types of the words and would not be in touch with the length of the specific words that are chosen for our tweet. We decided to approximate this constraint to a limited number of words in our tweet.

#yolo

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.

3.1 Implementation (0–2 pages)

The implementation relied on a Part-of-Speech tagger from Stanford's Natural Language Processsing group.

4 Experimental results (1–4 pages)

Some images here and stuff would be nice.

4.1 Experiemntal setup

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

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 4 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 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 bla bla

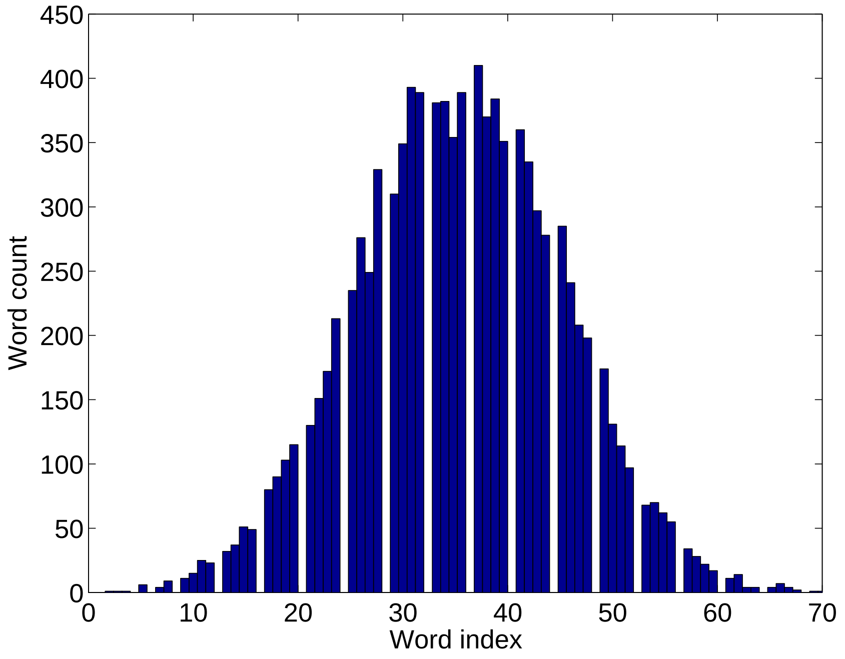


Figure 4: A description that makes browsing the paper easy and clearly describes what is in the picture. Make sure that the text in the figure is large enough to read and that the axes are labelled.

5 Summary and Conclusions (0.5–1 page)

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

References

Gabriele Barbieri, Franois Pachet, Pierre Roy, and Mirko Degli Esposti. Markov constraints for generating lyrics with style. 242:115–120, 2012.