Discourse

COMP90042

Natural Language Processing

Lecture 12

Semester 1 2022 Week 6

Jey Han Lau

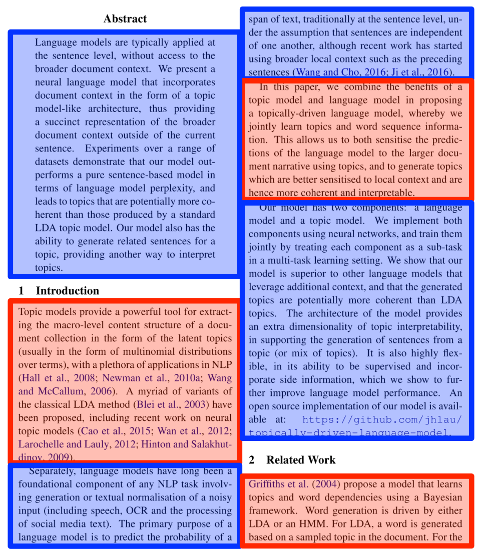
THE UNIVERSITY OF MELBOURNE

Discourse

- Most tasks/models we learned operate at word or sentence level:
  - POS tagging
  - Language models
  - Lexical/distributional semantics
- But NLP often deals with documents
- **Discourse**: understanding how sentences relate to each other in a document

Outline

- Discourse segmentation



Abstract

Language models are typically applied at the sentence level, without access to the broader document context. We present a neural language model that incorporates document context in the form of a topic model-like architecture, thus providing a succinct representation of the broader document context outside of the current sentence. Experiments over a range of datasets demonstrate that our model outperforms a pure sentence-based model in terms of language model perplexity, and leads to topics that are potentially more coherent than those produced by a standard LDA topic model. Our model also has the ability to generate related sentences for a topic, providing another way to interpret topics.

1 Introduction

Topic models provide a powerful tool for extracting the macro-level content structure of a document collection in the form of the latent topics (usually in the form of multinomial distributions over terms), with a plethora of applications in NLP (Hall et al., 2008; Newman et al., 2010a; Wang and McCallum, 2006). A myriad of variants of the classical LDA method (Blei et al., 2003) have been proposed, including recent work on neural topic models (Cao et al., 2015; Wan et al., 2012; Larochelle and Lauly, 2012; Hinton and Salakhutdinov, 2009).

Separately, language models have long been a foundational component of any NLP task involving generation or textual normalisation of a noisy input (including speech, OCR and the processing of social media text). The primary purpose of a language model is to predict the probability of a span of text, traditionally at the sentence level, under the assumption that sentences are independent of one another, although recent work has started using broader local context such as the preceding sentences (Wang and Cho, 2016; Ji et al., 2016).

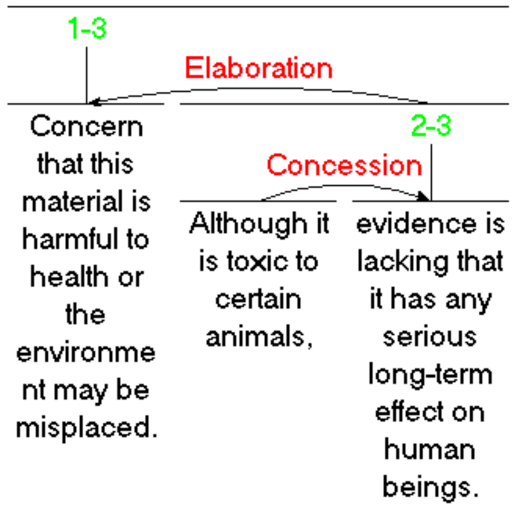
In this paper, we combine the benefits of a topic model and language model in proposing a topically-driven language model, whereby we jointly learn topics and word sequence information. This allows us to both sensitise the predictions of the language model to the larger document narrative using topics, and to generate topics which are better sensitised to local context and are hence more coherent and interpretable.

Our model has two components: a language model and a topic model. We implement both components using neural networks, and train them jointly by treating each component as a sub-task in a multi-task learning setting. We show that our model is superior to other language models that leverage additional context, and that the generated topics are potentially more coherent than LDA topics. The architecture of the model provides an extra dimensionality of topic interpretability, in supporting the generation of sentences from a topic (or mix of topics). It is also highly flexible, in its ability to be supervised and incorporate side information, which we show to further improve language model performance. An open source implementation of our model is available at: https://github.com/jhlau/topically-driven-language-model.

2 Related Work

Griffiths et al. (2004) propose a model that learns topics and word dependencies using a Bayesian framework. Word generation is driven by either LDA or an HMM. For LDA, a word is generated based on a sampled topic in the document. For the

- Discourse parsing

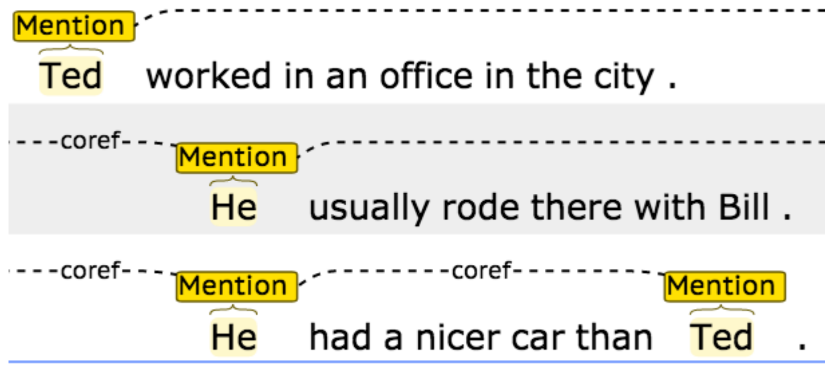


Concern that this material is harmful to health or the environment may be misplaced.

Although it is toxic to certain animals,

evidence is lacking that it has any serious long-term effect on human beings.

- Anaphora resolution



Ted worked in an office in the city .

He usually rode there with Bill .

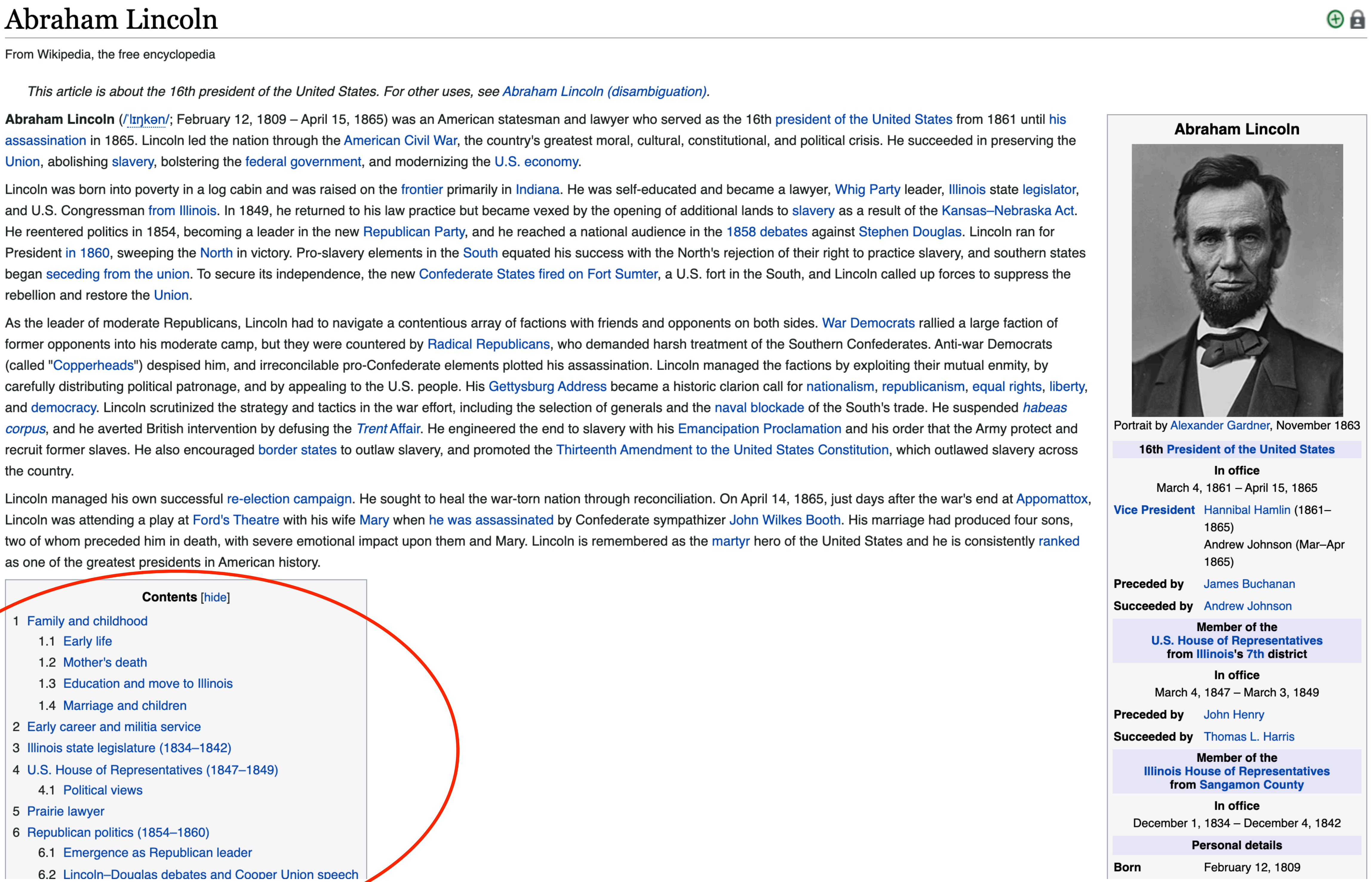
He had a nicer car than Ted .

Discourse Segmentation

Discourse Segmentation

- A document can be viewed as a sequence of segments
- A segment: a span of cohesive text
- Cohesion: organised around a **topic** or **function**

- Wikipedia biographies: early years, major events, impact on others



Abraham Lincoln

From Wikipedia, the free encyclopedia

This article is about the 16th president of the United States. For other uses, see Abraham Lincoln (disambiguation).

Abraham Lincoln (/ˈlɪŋkən/; February 12, 1809 – April 15, 1865) was an American statesman and lawyer who served as the 16th president of the United States from 1861 until his assassination in 1865. Lincoln led the nation through the American Civil War, the country's greatest moral, cultural, constitutional, and political crisis. He succeeded in preserving the Union, abolishing slavery, bolstering the federal government, and modernizing the U.S. economy.

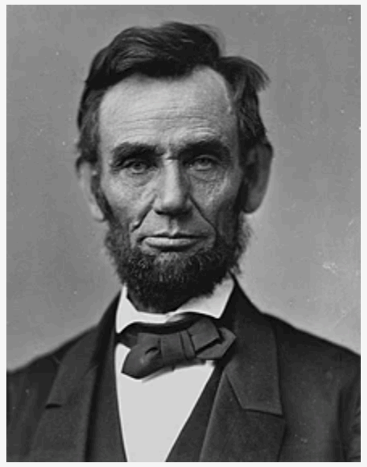
Lincoln was born into poverty in a log cabin and was raised on the frontier primarily in Indiana. He was self-educated and became a lawyer, Whig Party leader, Illinois state legislator, and U.S. Congressman from Illinois. In 1849, he returned to his law practice but became vexed by the opening of additional lands to slavery as a result of the Kansas–Nebraska Act. He reentered politics in 1854, becoming a leader in the new Republican Party, and he reached a national audience in the 1858 debates against Stephen Douglas. Lincoln ran for President in 1860, sweeping the North in victory. Pro-slavery elements in the South equated his success with the North's rejection of their right to practice slavery, and southern states began seceding from the union. To secure its independence, the new Confederate States fired on Fort Sumter, a U.S. fort in the South, and Lincoln called up forces to suppress the rebellion and restore the Union.

As the leader of moderate Republicans, Lincoln had to navigate a contentious array of factions with friends and opponents on both sides. War Democrats rallied a large faction of former opponents into his moderate camp, but they were countered by Radical Republicans, who demanded harsh treatment of the Southern Confederates. Anti-war Democrats (called "Copperheads") despised him, and irreconcilable pro-Confederate elements plotted his assassination. Lincoln managed the factions by exploiting their mutual enmity, by carefully distributing political patronage, and by appealing to the U.S. people. His Gettysburg Address became a historic clarion call for nationalism, republicanism, equal rights, liberty, and democracy. Lincoln scrutinized the strategy and tactics in the war effort, including the selection of generals and the naval blockade of the South's trade. He suspended *habeas corpus*, and he averted British intervention by defusing the *Trent* Affair. He engineered the end to slavery with his Emancipation Proclamation and his order that the Army protect and recruit former slaves. He also encouraged border states to outlaw slavery, and promoted the Thirteenth Amendment to the United States Constitution, which outlawed slavery across the country.

Lincoln managed his own successful re-election campaign. He sought to heal the war-torn nation through reconciliation. On April 14, 1865, just days after the war's end at Appomattox, Lincoln was attending a play at Ford's Theatre with his wife Mary when he was assassinated by Confederate sympathizer John Wilkes Booth. His marriage had produced four sons, two of whom preceded him in death, with severe emotional impact upon them and Mary. Lincoln is remembered as the martyr hero of the United States and he is consistently ranked as one of the greatest presidents in American history.

Contents [hide]

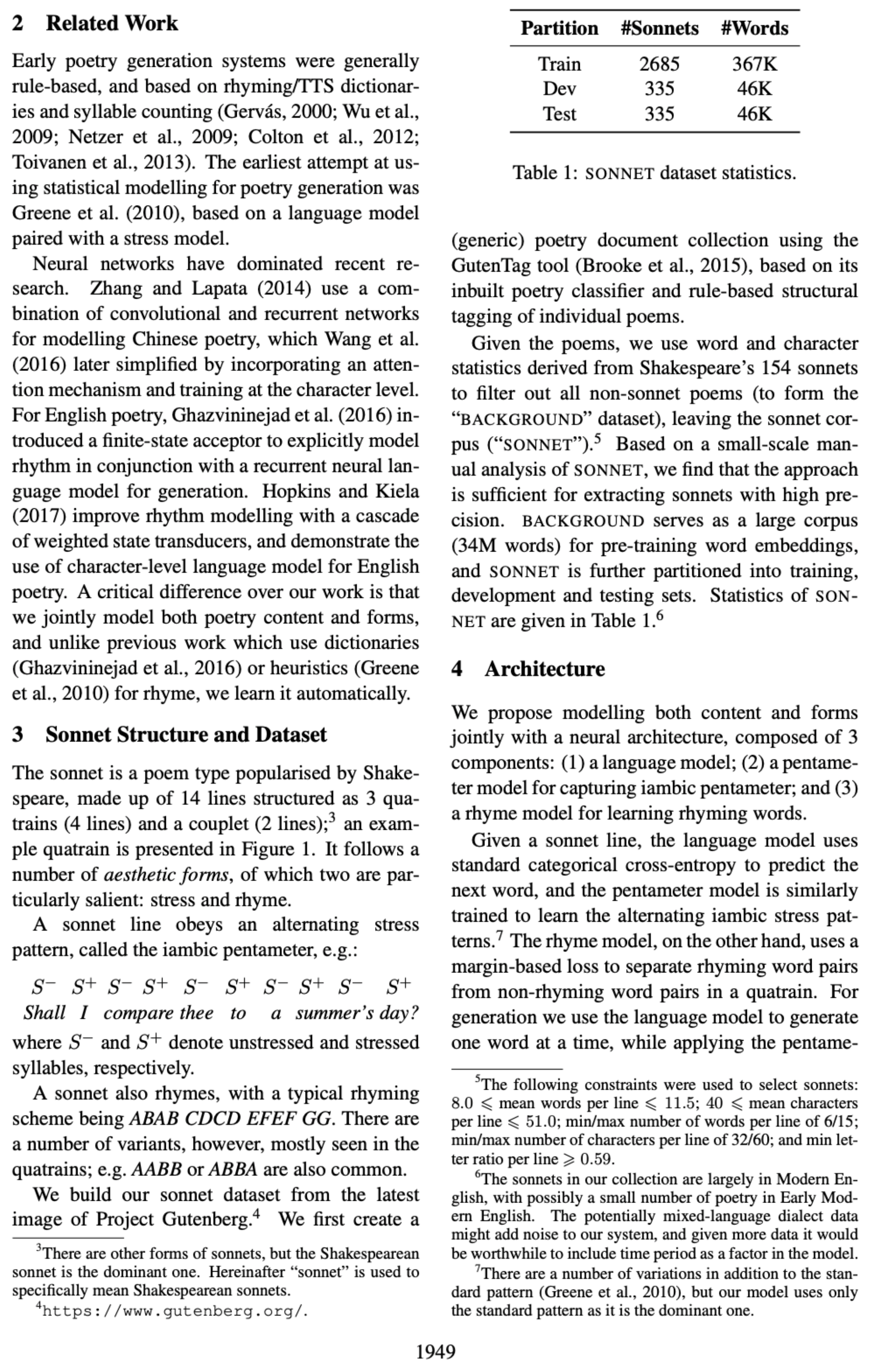
Abraham Lincoln



Portrait by Alexander Gardner, November 1863

16th President of the United States	
In office March 4, 1861 – April 15, 1865	
Vice President	Hannibal Hamlin (1861–1865) Andrew Johnson (Mar–Apr 1865)
Preceded by	James Buchanan
Succeeded by	Andrew Johnson
Member of the U.S. House of Representatives from Illinois's 7th district	
In office March 4, 1847 – March 3, 1849	
Preceded by	John Henry
Succeeded by	Thomas L. Harris
Member of the Illinois House of Representatives from Sangamon County	
In office December 1, 1834 – December 4, 1842	
Personal details	
Born	February 12, 1809

- Scientific articles: introduction, related work, experiments



2 Related Work

Early poetry generation systems were generally rule-based, and based on rhyming/TTS dictionaries and syllable counting (Gervás, 2000; Wu et al., 2009; Netzer et al., 2009; Colton et al., 2012; Toivanen et al., 2013). The earliest attempt at using statistical modelling for poetry generation was Greene et al. (2010), based on a language model paired with a stress model.

Neural networks have dominated recent research. Zhang and Lapata (2014) use a combination of convolutional and recurrent networks for modelling Chinese poetry, which Wang et al. (2016) later simplified by incorporating an attention mechanism and training at the character level. For English poetry, Ghazvininejad et al. (2016) introduced a finite-state acceptor to explicitly model rhythm in conjunction with a recurrent neural language model for generation. Hopkins and Kiela (2017) improve rhythm modelling with a cascade of weighted state transducers, and demonstrate the use of character-level language model for English poetry. A critical difference over our work is that we jointly model both poetry content and forms, and unlike previous work which use dictionaries (Ghazvininejad et al., 2016) or heuristics (Greene et al., 2010) for rhyme, we learn it automatically.

3 Sonnet Structure and Dataset

The sonnet is a poem type popularised by Shakespeare, made up of 14 lines structured as 3 quatrains (4 lines) and a couplet (2 lines);[3] an example quatrain is presented in Figure 1. It follows a number of *aesthetic forms*, of which two are particularly salient: stress and rhyme.

A sonnet line obeys an alternating stress pattern, called the iambic pentameter, e.g.:

$$\begin{array}{cccccccccc} S^- & S^+ & S^- & S^+ & S^- & S^+ & S^- & S^+ & S^- & S^+ \\ \textit{Shall} & \textit{I} & \textit{compare} & & \textit{thee} & \textit{to} & \textit{a} & & \textit{summer's} & \textit{day?} \end{array}$$

where S^- and S^+ denote unstressed and stressed syllables, respectively.

A sonnet also rhymes, with a typical rhyming scheme being *ABAB CDCD EFEF GG*. There are a number of variants, however, mostly seen in the quatrains; e.g. *AABB* or *ABBA* are also common.

We build our sonnet dataset from the latest image of Project Gutenberg.[4] We first create a (generic) poetry document collection using the GutenTag tool (Brooke et al., 2015), based on its inbuilt poetry classifier and rule-based structural tagging of individual poems.

Given the poems, we use word and character statistics derived from Shakespeare's 154 sonnets to filter out all non-sonnet poems (to form the "BACKGROUND" dataset), leaving the sonnet corpus ("SONNET").[5] Based on a small-scale manual analysis of SONNET, we find that the approach is sufficient for extracting sonnets with high precision. BACKGROUND serves as a large corpus (34M words) for pre-training word embeddings, and SONNET is further partitioned into training, development and testing sets. Statistics of SONNET are given in Table 1.[6]

Partition	#Sonnets	#Words
Train	2685	367K
Dev	335	46K
Test	335	46K

Table 1: SONNET dataset statistics.

4 Architecture

We propose modelling both content and forms jointly with a neural architecture, composed of 3 components: (1) a language model; (2) a pentameter model for capturing iambic pentameter; and (3) a rhyme model for learning rhyming words.

Given a sonnet line, the language model uses standard categorical cross-entropy to predict the next word, and the pentameter model is similarly trained to learn the alternating iambic stress patterns.[7] The rhyme model, on the other hand, uses a margin-based loss to separate rhyming word pairs from non-rhyming word pairs in a quatrain. For generation we use the language model to generate one word at a time, while applying the pentame-

[3]There are other forms of sonnets, but the Shakespearean sonnet is the dominant one. Hereinafter "sonnet" is used to specifically mean Shakespearean sonnets.

[4]`https://www.gutenberg.org/`.

[5]The following constraints were used to select sonnets: $8.0 \leqslant$ mean words per line $\leqslant 11.5$; $40 \leqslant$ mean characters per line $\leqslant 51.0$; min/max number of words per line of 6/15; min/max number of characters per line of 32/60; and min letter ratio per line $\geqslant 0.59$.

[6]The sonnets in our collection are largely in Modern English, with possibly a small number of poetry in Early Modern English. The potentially mixed-language dialect data might add noise to our system, and given more data it would be worthwhile to include time period as a factor in the model.

[7]There are a number of variations in addition to the standard pattern (Greene et al., 2010), but our model uses only the standard pattern as it is the dominant one.

1949

Unsupervised Approaches

- TextTiling algorithm: looking for points of low lexical cohesion between sentences
- For each sentence gap:
  - Create two BOW vectors consisting of words from *k* sentences on either side of gap
  - Use cosine to get a similarity score (*sim*) for two vectors
  - For gap *i,* calculate a depth score, insert boundaries when depth is greater than some threshold *t*

$$depth(gap_i) = (sim_{i-1} - sim_i) + (sim_{i+1} - sim_i)$$

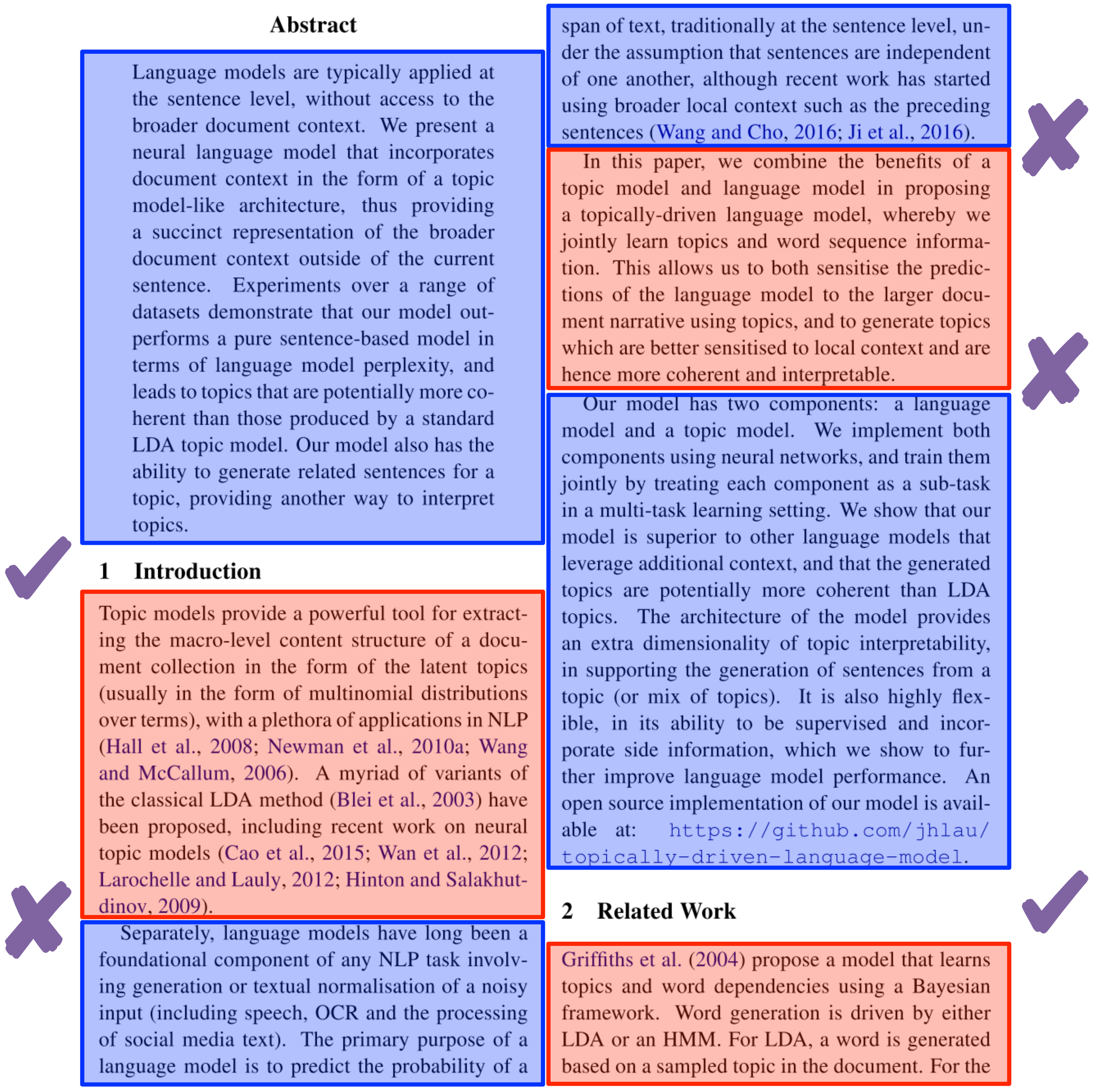
Text Tiling Example (k=1, t=0.9)

		He walked 15 minutes to the tram stop.
d=0.7-0.9=-0.2	sim: 0.9	
		Then he waited for another 20 minutes, but the tram didn't come.
d=(0.9-0.7)+(0.1-0.7)=-0.4	sim: 0.7	
		The tram drivers were on strike that morning.
d=(0.7-0.1)+(0.5-0.1)=1.0	sim: 0.1	
		So he walked home and got his bike out of the garage.
d=(0.1-0.5)+(0.8-0.5)=-0.1	sim: 0.5	
		He started riding but quickly discovered he had a flat tire
d=(0.1-0.5)+(0.8-0.5)=-0.6	sim: 0.8	
		He walked his bike back home.
d=0.8-0.5=0.3	sim: 0.5	
		He looked around but his wife had cleaned the garage and he couldn't find the bike pump.

$$depth(gap_i) = (sim_{i-1} - sim_i) + (sim_{i+1} - sim_i)$$

Supervised Approaches

- Get labelled data from easy sources
  - Scientific publications
  - Wikipedia articles



Abstract

Language models are typically applied at the sentence level, without access to the broader document context. We present a neural language model that incorporates document context in the form of a topic model-like architecture, thus providing a succinct representation of the broader document context outside of the current sentence. Experiments over a range of datasets demonstrate that our model outperforms a pure sentence-based model in terms of language model perplexity, and leads to topics that are potentially more coherent than those produced by a standard LDA topic model. Our model also has the ability to generate related sentences for a topic, providing another way to interpret topics.

1 Introduction

Topic models provide a powerful tool for extracting the macro-level content structure of a document collection in the form of the latent topics (usually in the form of multinomial distributions over terms), with a plethora of applications in NLP (Hall et al., 2008; Newman et al., 2010a; Wang and McCallum, 2006). A myriad of variants of the classical LDA method (Blei et al., 2003) have been proposed, including recent work on neural topic models (Cao et al., 2015; Wan et al., 2012; Larochelle and Lauly, 2012; Hinton and Salakhutdinov, 2009).

Separately, language models have long been a foundational component of any NLP task involving generation or textual normalisation of a noisy input (including speech, OCR and the processing of social media text). The primary purpose of a language model is to predict the probability of a span of text, traditionally at the sentence level, under the assumption that sentences are independent of one another, although recent work has started using broader local context such as the preceding sentences (Wang and Cho, 2016; Ji et al., 2016).

In this paper, we combine the benefits of a topic model and language model in proposing a topically-driven language model, whereby we jointly learn topics and word sequence information. This allows us to both sensitise the predictions of the language model to the larger document narrative using topics, and to generate topics which are better sensitised to local context and are hence more coherent and interpretable.

Our model has two components: a language model and a topic model. We implement both components using neural networks, and train them jointly by treating each component as a sub-task in a multi-task learning setting. We show that our model is superior to other language models that leverage additional context, and that the generated topics are potentially more coherent than LDA topics. The architecture of the model provides an extra dimensionality of topic interpretability, in supporting the generation of sentences from a topic (or mix of topics). It is also highly flexible, in its ability to be supervised and incorporate side information, which we show to further improve language model performance. An open source implementation of our model is available at: https://github.com/jhlau/topically-driven-language-model.

2 Related Work

Griffiths et al. (2004) propose a model that learns topics and word dependencies using a Bayesian framework. Word generation is driven by either LDA or an HMM. For LDA, a word is generated based on a sampled topic in the document. For the

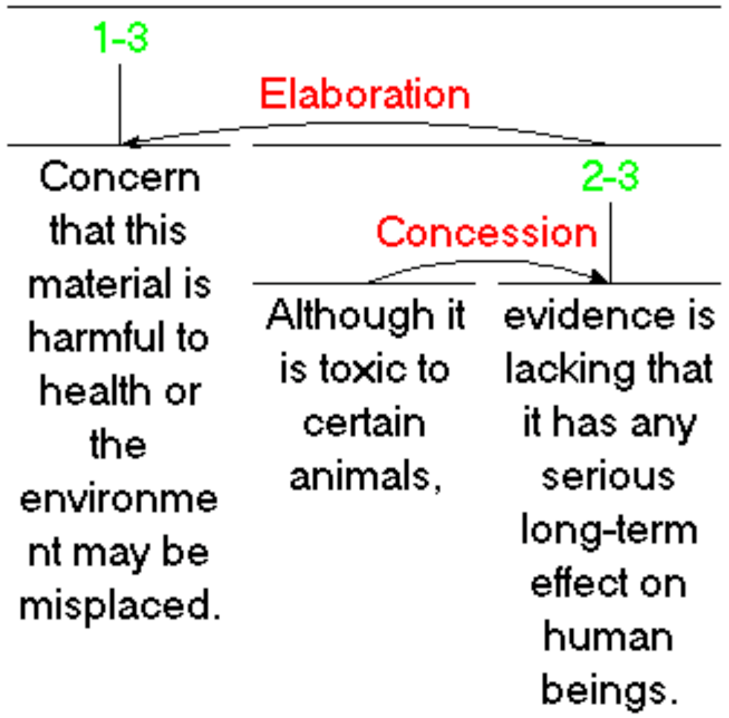
Supervised Discourse Segmenter

- Apply a binary classifier to identify boundaries
- Or use sequential classifiers
- Potentially include classification of section types (introduction, conclusion, etc.)
- Integrate a wider range of features, including
  - distributional semantics
  - discourse markers (*therefore*, *and*, etc)

Discourse Parsing

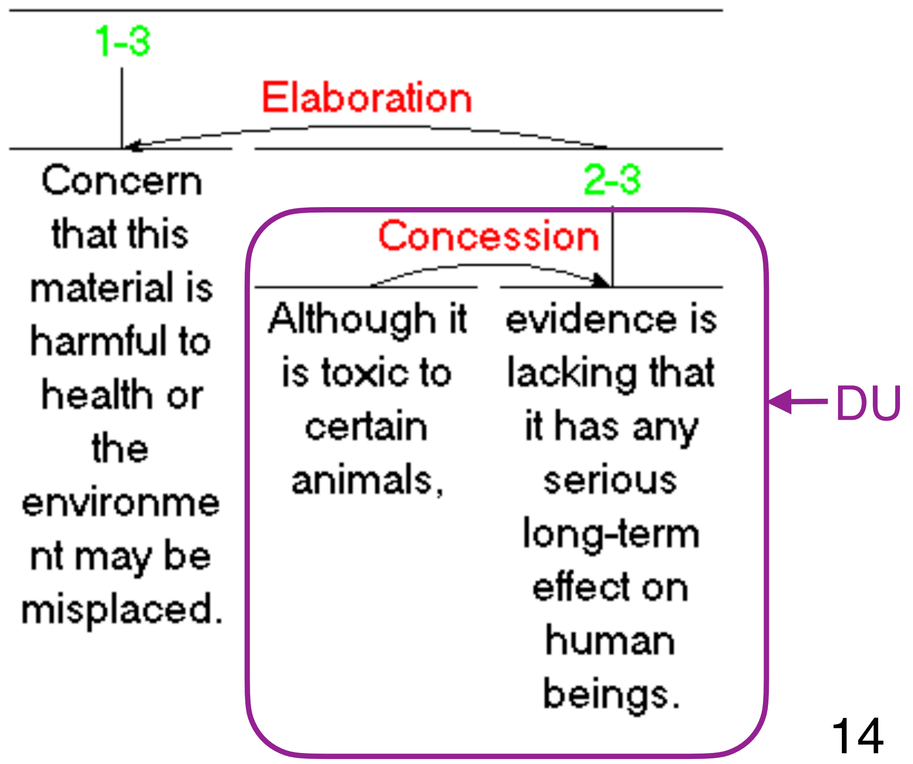
Discourse Analysis

- Identify **discourse units**, and the **relations** that hold between them
- **Rhetorical Structure Theory (RST)** is a framework to do hierarchical analysis of discourse structure in documents


1-3
Elaboration
Concern that this material is harmful to health or the environment may be misplaced.
2-3
Concession
Although it is toxic to certain animals,
evidence is lacking that it has any serious long-term effect on human beings.

Discourse Units

- Typically clauses of a sentence
- DUs do not cross sentence boundary
- *[It does have beautiful scenery,] [some of the best since Lord of the Rings.]*
- 2 merged DUs = another composite DU


1-3
Elaboration
Concern that this material is harmful to health or the environment may be misplaced.
2-3
Concession
Although it is toxic to certain animals,
evidence is lacking that it has any serious long-term effect on human beings.
DU

Discouse Relations

- **Relations** between discourse units:
  - *conjuction*, *justify*, *concession*, *elaboration*, etc
  - *[It does have beautiful scenery,]*

    ↑*(elaboration)*

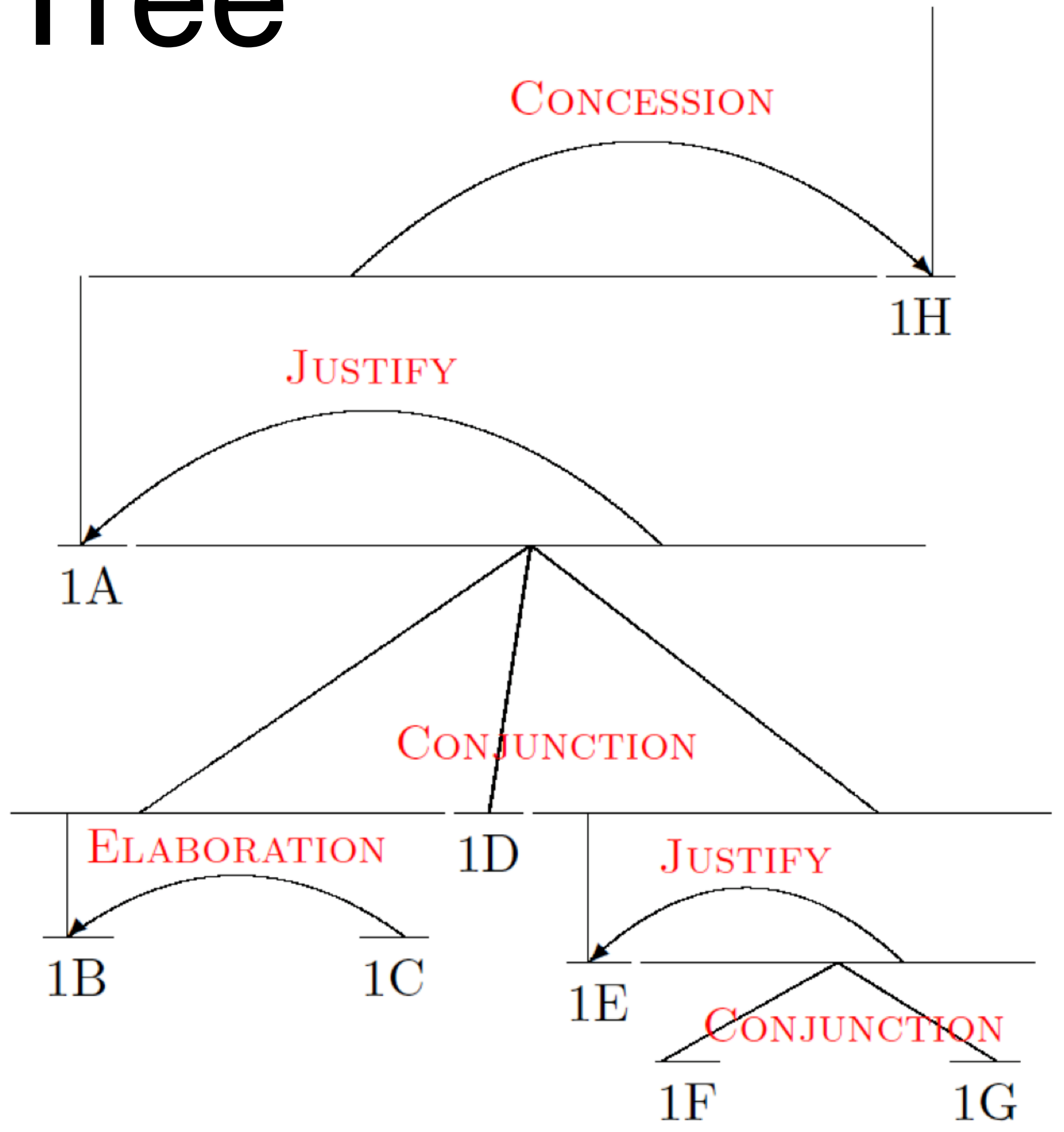
    [some of the best since Lord of the Rings.]

Nucleus vs. Satellite

- Within a discourse relation, one argument is the **nucleus** (the primary argument)
- The supporting argument is the **satellite**
  - *[It does have beautiful scenery,]$_{nucleus}$*
    $\uparrow$ *(elaboration)*
    [some of the best since Lord of the Rings.]$_{satellite}$
- Some relations are equal (e.g. conjunction), and so both arguments are nuclei
  - *[He was a likable chap,]$_{nucleus}$*
    $\uparrow$ *(conjunction)*
    [and I hated to see him die.]$_{nucleus}$

RST Tree

- An RST relation combines two or more DUs into composite DUs
- Process of combining DUs is repeated, creating an RST tree

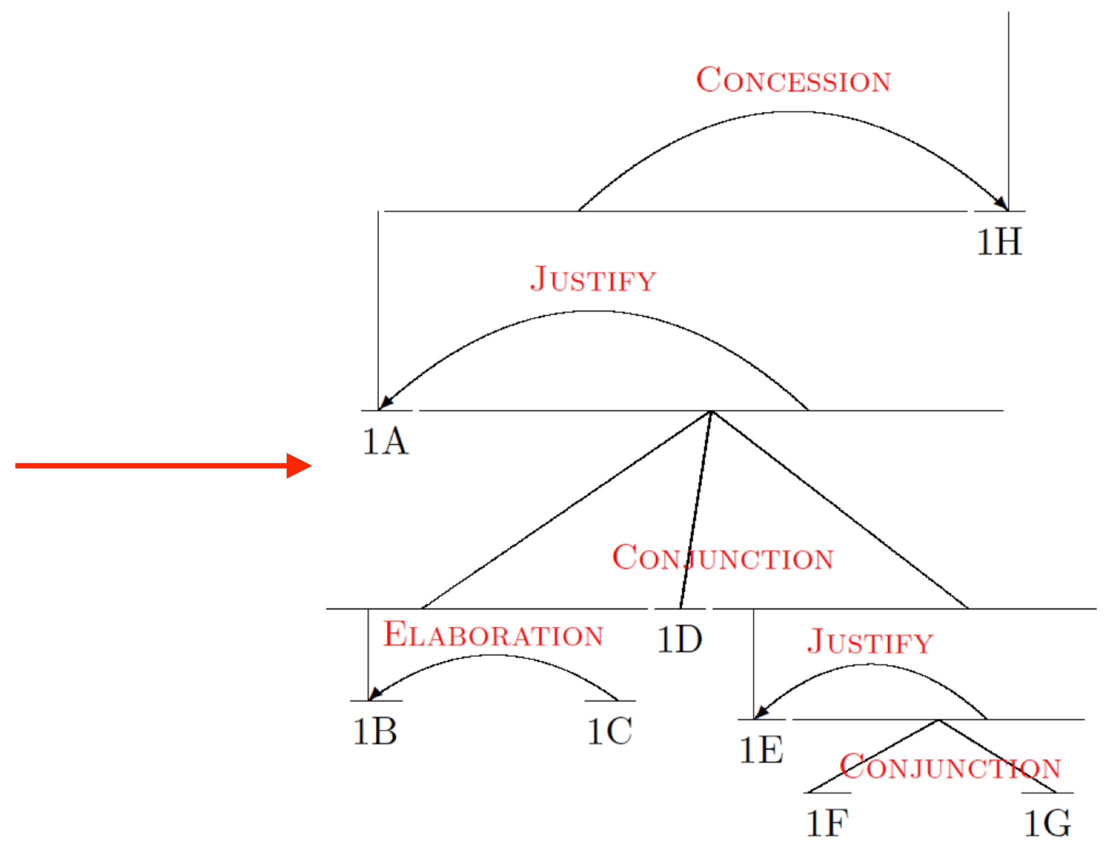


1A:	[It could have been a great movie.]
1B:	[It does have beautiful scenery,]
1C:	[some of the best since Lord of the Rings.]
1D:	[The acting is well done,]
1E:	[and I really liked the son of the leader of the Samurai.]
1F:	[He was a likable chap,]
1G:	[and I hated to see him die.]
1H:	[But, other than all that, this movie is nothing more than hidden rip-offs.]

RST Parsing

- Task: given a document, recover the RST tree
    - Rule-basd parsing
    - Bottom-up approach
    - Top-down aproach

1A:	[It could have been a great movie.]
1B:	[It does have beautiful scenery,]
1C:	[some of the best since Lord of the Rings.]
1D:	[The acting is well done,]
1E:	[and I really liked the son of the leader of the Samurai.]
1F:	[He was a likable chap,]
1G:	[and I hated to see him die.]
1H:	[But, other than all that, this movie is nothing more than hidden rip-offs.]



Parsing Using Discourse Markers

- Some discourse markers (cue phrases) explicitly indicate relations
  - *although*, *but*, *for example*, *in other words*, *so*, *because, in conclusion*,…
- Can be used to build a simple rule-based parser
- However
  - Many relations are not marked by discourse marker
  - Many discourse markers ambiguous (e.g. *and*)

Parsing Using Machine Learning

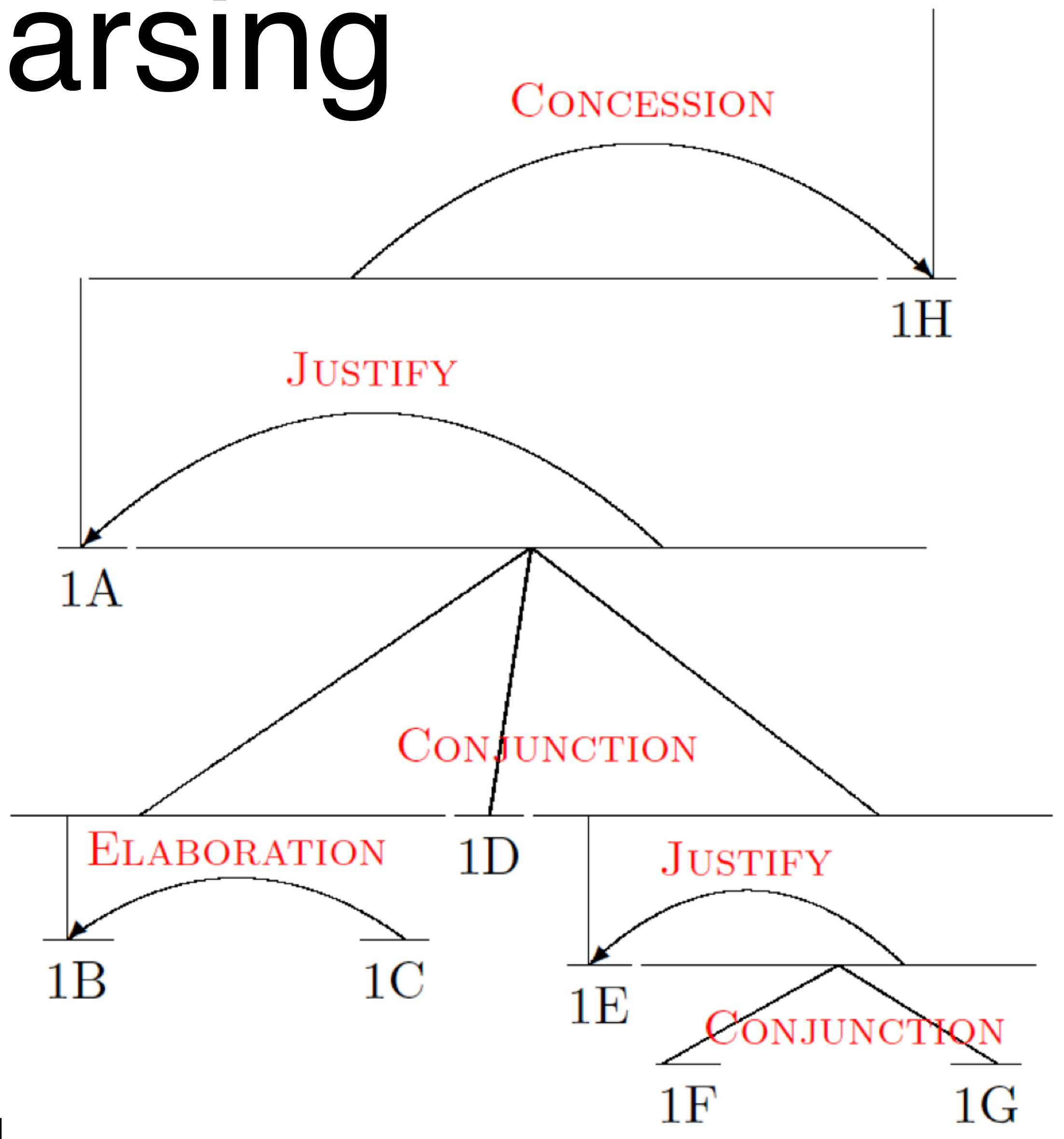
- RST Discourse Treebank
  - 300+ documents annotated with RST trees
- Basic idea:
  - Segment document into DUs
  - Combine adjacent DUs into composite DUs iteratively to create the full RST tree (bottom-up parsing)

Bottom-Up Parsing

- Transition-based parsing (lecture 16):
  - Greedy, uses shift-reduce algorithm
- CYK/chart parsing algorithm (lecture 14)
  - Global, but some constraints prevent CYK from finding globally optimal tree for discourse parsing

Top-Down Parsing

1. Segment document into DUs
2. Decide a boundary to split into 2 segments
3. For each segment, repeat step 2



1A: [It could have been a great movie.]
2 →
1B: [It does have beautiful scenery,]
4 →
1C: [some of the best since Lord of the Rings.]
3 →
1D: [The acting is well done,]
5 →
1E: [and I really liked the son of the leader of the Samurai.]
6 →
1F: [He was a likable chap,]
7 →
1G: [and I hated to see him die.]
1 →
1H: [But, other than all that, this movie is nothing more than hidden rip-offs.]

Discourse Parsing Features

- Bag of words
- Discourse markers
- Starting/ending *n*-grams
- Location in the text
- Syntax features
- Lexical and distributional similarities

Applications of Discourse Parsing?

- Summarisation
- Sentiment analysis
- Argumentation
- Authorship attribution
- Essay scoring

PollEv.com/jeyhanlau569

Anaphora Resolution

Anaphors

- **Anaphor**: linguistic expressions that refer back to earlier elements in the text
- Anaphors have a **antecedent** in the discourse, often but not always a noun phrase
  - *Yesterday, Ted was late for work. It all started when his car wouldn't start.*
- Pronouns are the most common anaphor
- But there are various others
  - Demonstratives (*that problem*)

Motivation

- Essential for deep semantic analysis
    - Very useful for QA, e.g., reading comprehension

Ted's car broke down. So he went over to Bill's house to borrow his car. Bill said that was fine.

Whose car is borrowed?

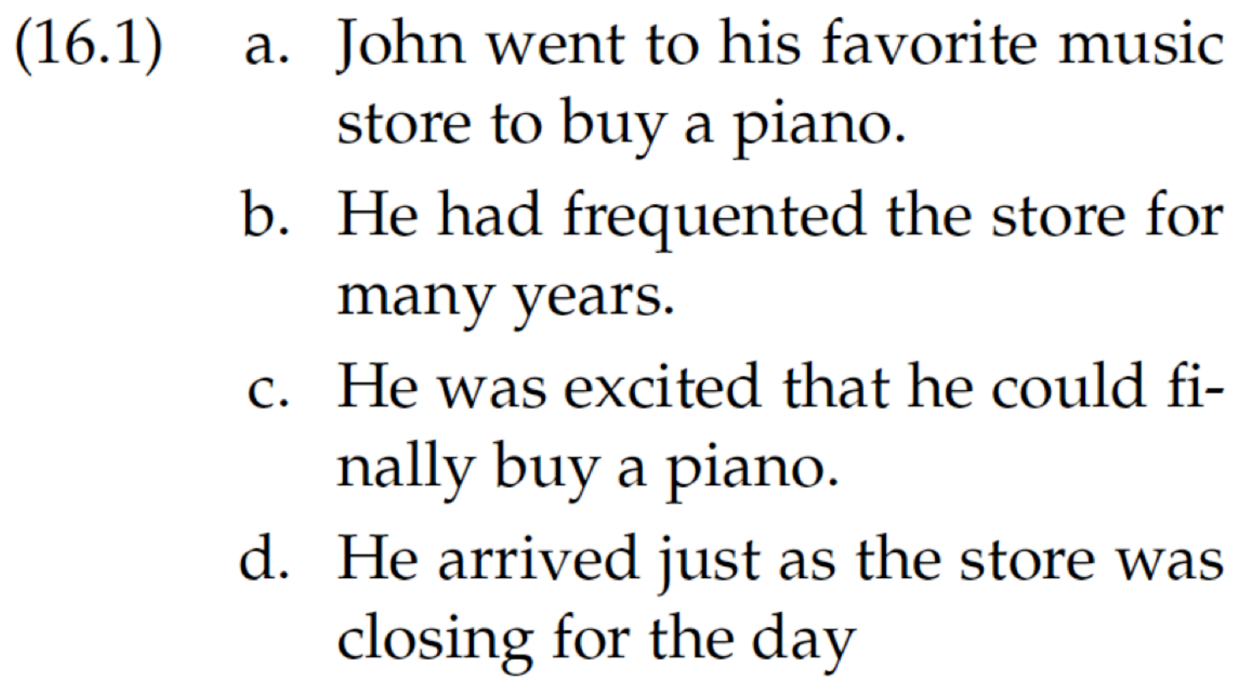
Antecedent Restrictions

- Pronouns must agree in **number** with their antecedents
  - *His coworkers were leaving for lunch when Ted arrived. They invited him, but he said no.*
- Pronouns must agree in **gender** with their antecedents
  - *Sue was leaving for lunch when Ted arrived. She invited him, but he said no.*
- Pronouns whose antecedents are the subject of the same syntactic clause must be **reflexive** *(…self)*
  - *Ted was angry at him.* [*him ≠ Ted*]
  - *Ted was angry at himself.* [*himself = Ted*]

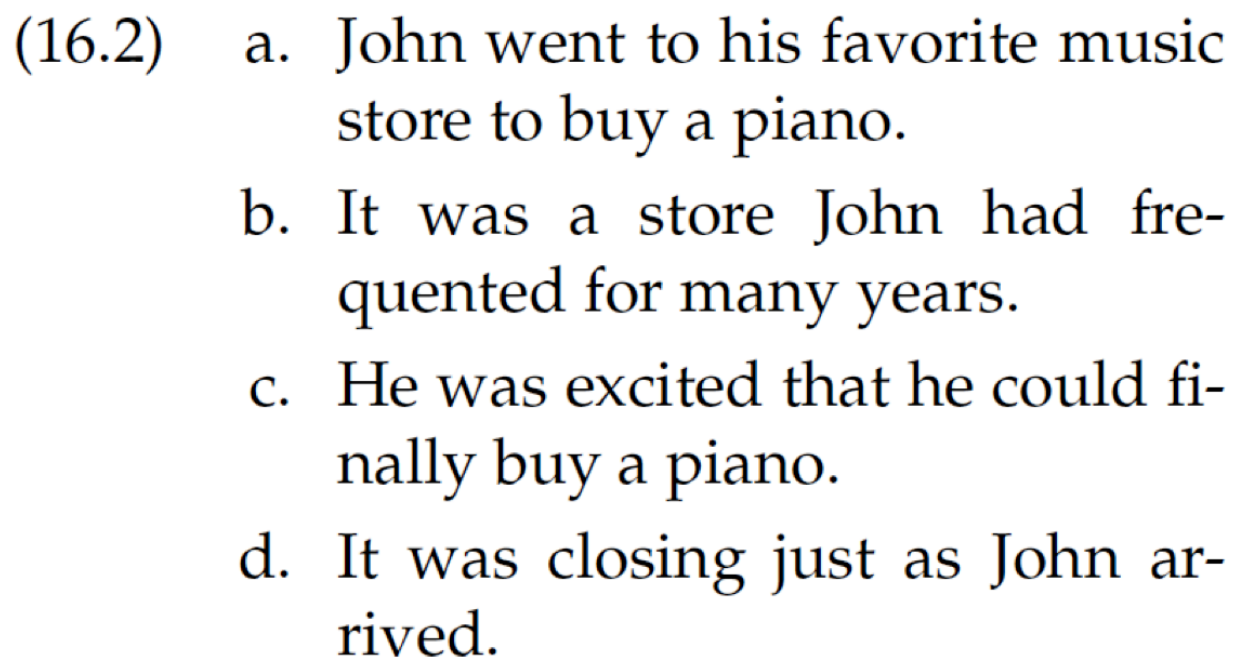
Antecedent Preferences

- The antecedents of pronouns should be recent
  - *He waited for another 20 minutes, but the tram didn't come. So he walked home and got his bike out of the garage. He started riding it to work.*
- The antecedent should be salient, as determined by grammatical position
  - Subject > object > argument of preposition
  - *Ted usually rode to work with Bill. He was never late.*

Entities and Reference



(16.1)
a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day



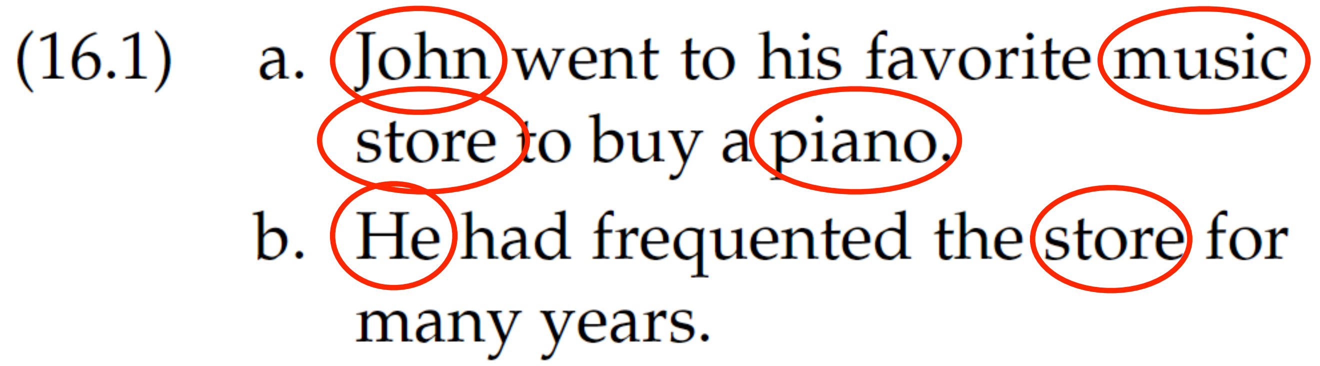
(16.2)
a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

- Discourse 16.1 (left) more coherent
- Pronouns all refer to John consistently, the protagonist

Centering Theory

- A unified account of relationship between discourse structure and entity reference
- Every utterance in the discourse is characterised by a set of entities, known as **centers**
- Explains preference of certain entities for ambiguous pronouns

For an Utterance U_n



(16.1) a. John went to his favorite music store to buy a piano.

b. He had frequented the store for many years.

- **Forward-looking centers**:
  - All entities in U_n:
    $C_f(U_n) = [e_1, e_2, \ldots]$
  - C_f(16.1a) = [*John*, *music store*, *piano*]
  - Ordered by syntactic prominence: subjects > objects
- **Backward-looking center**:
  - Highest ranked forward-looking center in previous utterance ($C_f(U_{n-1})$) that is also in current utterance (U_n)
  - Candidate entities in 16.1b = [*John*, *music store*]
  - C_b(16.1b) = [*John*]
  - Not *music store* because *John* has a higher rank in previous utterance's forward-looking centers $C_f(U_{n-1})$

Centering Algorithm

- When resolving entity for anaphora resolution, choose the entity such that the **top foward-looking center** matches with the **backward-looking center**
- Why? Because the text reads more fluent when this condition is satisfied

(16.1)
a. John went to his favorite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day

(16.2)
a. John went to his favorite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

Text is coherent because the top forward-looking center matches the backward-looking center for each utterance:

top forward-looking center = *John*
backward-looking center = *John*

Not quite the case here.

C_f(16.2b) = [*music store*, *John*]
C_b(16.2b) = [*John*]

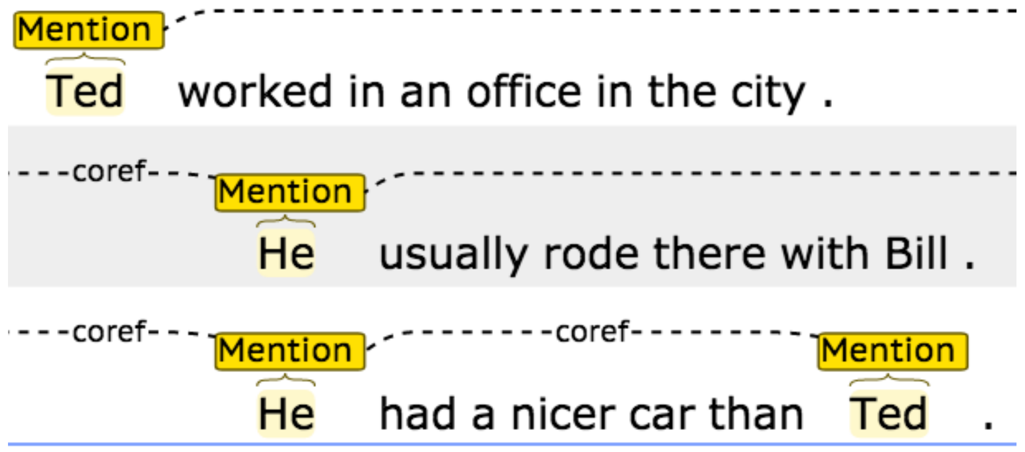
C_f(16.2d) = [*music store*, *John*]
C_b(16.2d) = [*John*]

Supervised Anaphor Resolution

- Build a binary classifier for anaphor/antecedent pairs
- Convert restrictions and preferences into features
  - Binary features for number/gender compatibility
  - Position of antecedent in text
  - Include features about type of antecedent
- With enough data, can approximate the centering algorithm
- But also easy to include features that are potentially helpful
  - words around anaphor/antecedent

Anaphora Resolution Tools

- Stanford CoreNLP includes pronoun coreference models
  - rule-based system isn't too bad
  - considerably faster than neural models



Ted worked in an office in the city .
He usually rode there with Bill .
He had a nicer car than Ted .

SYSTEM	LANGUAGE	PREPROCESSING TIME	COREF TIME	TOTAL TIME	F1 SCORE
Deterministic	English	3.87s	0.11s	3.98s	49.5
Statistical	English	0.48s	1.23s	1.71s	56.2
Neural	English	3.22s	4.96s	8.18s	60.0
Deterministic	Chinese	0.39s	0.16s	0.55s	47.5
Neural	Chinese	0.42s	7.02s	7.44s	53.9

Source: https://stanfordnlp.github.io/CoreNLP/coref.html
Evaluated on CoNLL 2012 task.

A Final Word

- For many tasks, it is important to consider context larger than sentences
- Traditionally many popular NLP applications has been sentence-focused (e.g. machine translation), but that is beginning to change…

Further Reading

- E18, Ch 16