IN4080 – 2020 FALL

NATURAL LANGUAGE PROCESSING

Jan Tore Lønning

IE: Relation extraction, encoder-decoders

Lecture 14, 16 Nov.

Today

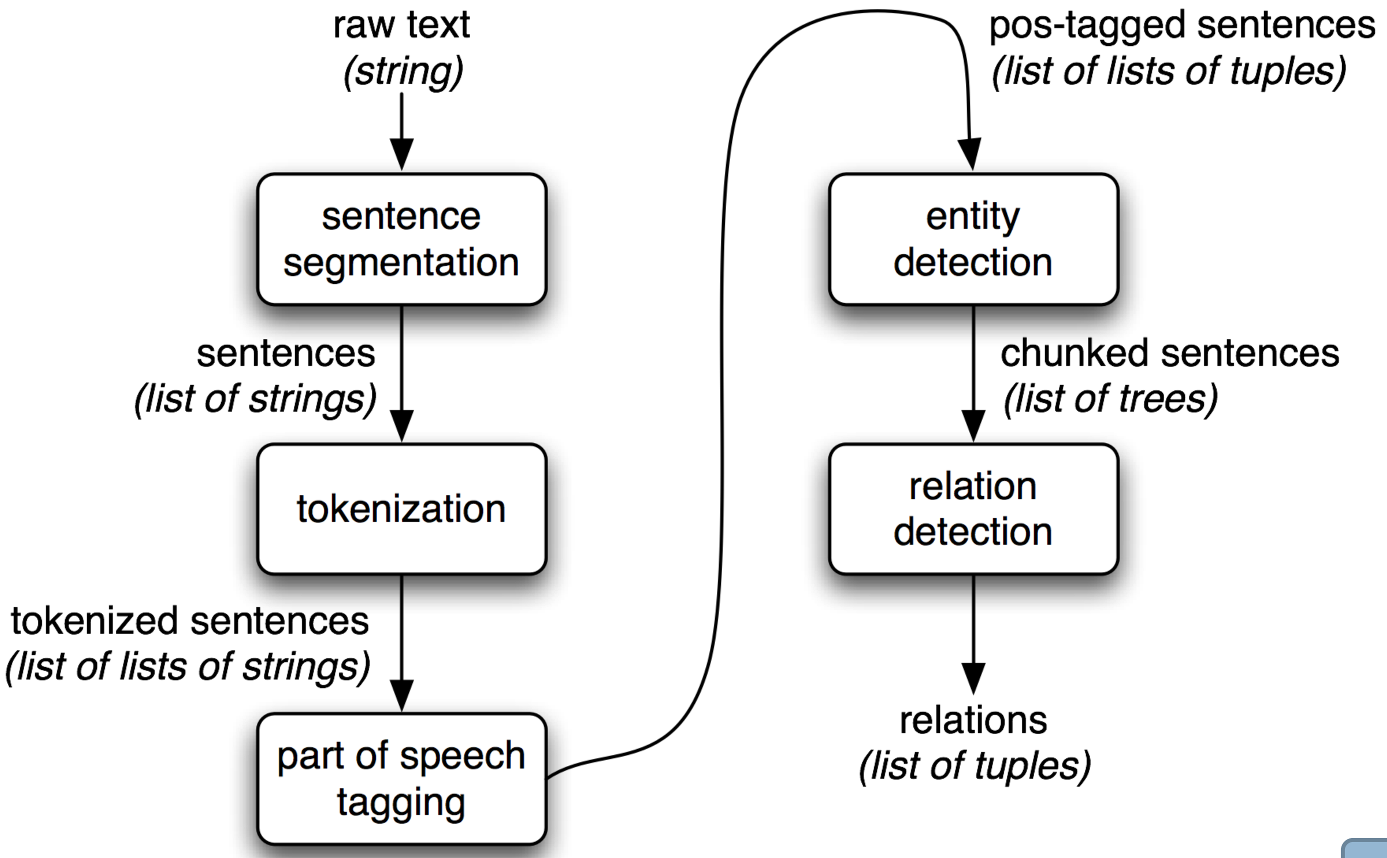
- Information extraction:
  - Relation extractions
    - 5 ways
- Two words on syntax
- Encoder-decoders
- Beam search

IE basics

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. (Wikipedia)

- Bottom-Up approach
- Start with unrestricted texts, and do the best you can
- The approach was in particular developed by the Message Understanding Conferences (MUC) in the 1990s
- Select a particular domain and task

A typical pipeline



From NLTK

Goal

- Extract the relations that exist between the (named) entities in the text
- A fixed set of relations (normally)
  - Determined by application:
    - Jeopardy
    - Preventing terrorist attacks
    - Detecting illness from medical record
    - …

- Born_in
- Date_of_birth
- Parent_of

- Author_of
- Winner_of

- Part_of
- Located_in

- Acquire
- Threaten

- Has_symptom
- Has_illness

Examples

Relations		Examples	Types
Affiliations			
	Personal	*married to, mother of*	PER $\longrightarrow$ PER
	Organizational	*spokesman for, president of*	PER $\longrightarrow$ ORG
	Artifactual	*owns, invented, produces*	(PER $\mid$ ORG) $\longrightarrow$ ART
Geospatial			
	Proximity	*near, on outskirts*	LOC $\longrightarrow$ LOC
	Directional	*southeast of*	LOC $\longrightarrow$ LOC
Part-Of			
	Organizational	*a unit of, parent of*	ORG $\longrightarrow$ ORG
	Political	*annexed, acquired*	GPE $\longrightarrow$ GPE

Today

- Information extraction:
  - Relation extractions
    - 5 ways
- Two words on syntax
- Encoder-decoders
- Beam search

Methods for relation extraction

1. Hand-written patterns
2. Machine Learning (Supervised classifiers)
3. Semi-supervised classifiers via bootstrapping
4. Semi-supervised classifiers via distant supervision
5. Unsupervised

1. Hand-written patterns

- Example: acquisitions
- [ORG]…(buy(s) | bought | aquire(s | d))…[ORG]

- Hand-write patterns like this
- Properties:
  - High precision
  - Will only cover a small set of patterns
  - Low recall
  - Time consuming
- (Also in NLTK, sec 7.6)

Example

NP {, NP}* {,} (and|or) other NP_H	temples, treasuries, and other important civic buildings
NP_H such as {NP,}* {(or|and)} NP	red algae such as Gelidium
such NP_H as {NP,}* {(or|and)} NP	such authors as Herrick, Goldsmith, and Shakespeare
NP_H {,} including {NP,}* {(or|and)} NP	common-law countries, including Canada and England
NP_H {,} especially {NP}* {(or|and)} NP	European countries, especially France, England, and Spain

Figure 17.12 Hand-built lexico-syntactic patterns for finding hypernyms, using {} to mark optionality (Hearst 1992a, Hearst 1998).

Methods for relation extraction

1. Hand-written patterns
2. Machine Learning (Supervised classifiers)
3. Semi-supervised classifiers via bootstrapping
4. Semi-supervised classifiers via distant supervision
5. Unsupervised

2. Supervised classifiers

- A corpus
- A fixed set of entities and relations
- The sentences in the corpus are hand-annotated:
  - Entities
  - Relations between them
- Split the corpus into parts for training and testing
- Train a classifier:
  - Choose learner:
    Naive Bayes, Logistic regression (Max Ent), SVM, …
  - Select features

2. Supervised classifiers, contd.

- Training:
  - Use pairs of entities within the same sentence with no relation between them as negative data
- Classification
  1. Find the NERs
  2. For each pair of NERs determine whether there is a relation between them
  3. If there is, label the relation

Examples of features

American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said

M1 headword	*airlines* (as a word token or an embedding)
M2 headword	*Wagner*
Word(s) before M1	NONE
Word(s) after M2	*said*
Bag of words between	{*a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman* }
M1 type	ORG
M2 type	PERS
Concatenated types	ORG-PERS
Constituent path	$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
Base phrase path	$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$
Typed-dependency path	$Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner$

Figure 17.14 Sample of features extracted during classification of the <American Airlines, Tim Wagner> tuple; M1 is the first mention, M2 the second.

Properties

- The bottleneck is the availability of training data
- To hand label data is time consuming
- Mostly applied to restricted domains
- Does not generalize well to other domains

Methods for relation extraction

1. Hand-written patterns
2. Machine Learning (Supervised classifiers)
3. Semi-supervised classifiers via bootstrapping
4. Semi-supervised classifiers via distant supervision
5. Unsupervised

3. Semisupervised, bootstrapping

Relation
ACQUIRE

Pairs:
IBM – AlchemyAPI
Google – YouTube
Facebook - WhatsApp

Patterns:
[ORG]…bought…[ORG]

- If we know a pattern for a relation, we can determine whether a pair stands in the relation
- Conversely: If we know that a pair stands in a relationship, we can find patterns that describe the relation

Example

- (IBM, AlchemyAPI): ACQUIRE
- Search for sentences containing IBM and AlchemyAPI
- Results (Web-search, Google, btw. first 10 results):
  - *IBM's Watson makes intelligent acquisition of Denver-based AlchemyAPI* (Denver Post)
  - *IBM is buying machine-learning systems maker AlchemyAPI Inc. to bolster its Watson technology as competition heats up in the data analytics and artificial intelligence fields.* (Bloomberg)
  - *IBM has acquired computing services provider AlchemyAPI to broaden its portfolio of Watson-branded cognitive computing services.* (ComputerWorld)

Example contd.

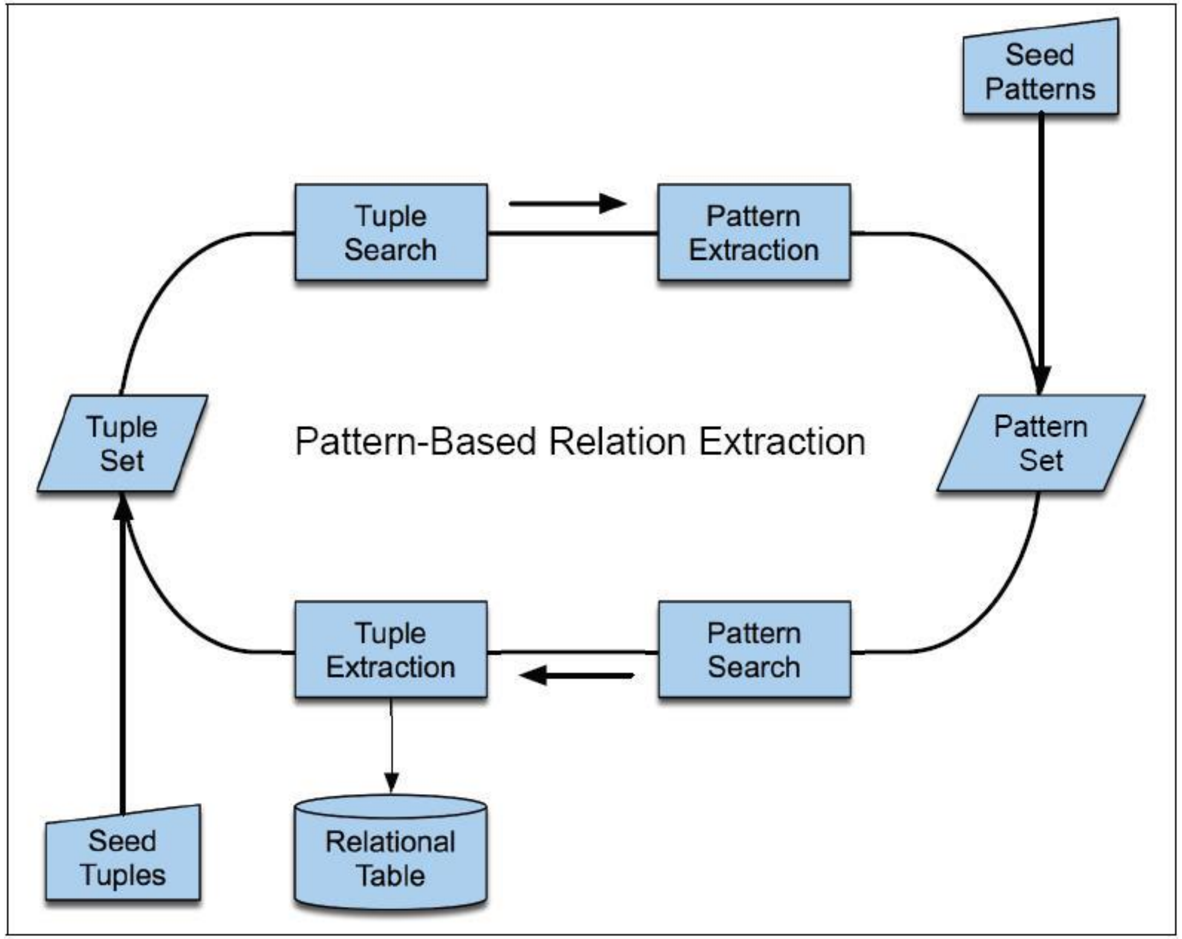
- Extract patterns
  - *IBM's Watson makes intelligent acquisition of Denver-based AlchemyAPI* (Denver Post)
  - *IBM is buying machine-learning systems maker AlchemyAPI Inc. to bolster its Watson technology as competition heats up in the data analytics and artificial intelligence fields.* (Bloomberg)
  - *IBM has acquired computing services provider AlchemyAPI to broaden its portfolio of Watson-branded cognitive computing services.* (ComputerWorld)

Procedure

- From the extracted sentences, we extract patterns
- Use these patterns to extract more pairs of entities that stand in these patterns
- These pairs may again be used for extracting more patterns, etc.

- …*makes intelligent acquisition* …
- … *is buying* …
- … *has acquired* …

Bootstrapping



A little more

- We could either
  - extract pattern templates and search for more occurrences of these patters in text, or
  - extract features for classification and build a classifier
- If we use patterns we should generalize
  - *makes intelligent acquisition* → *(make(s)|made) JJ* acquisition*
- During the process we should evaluate before we extend:
  - Does the new pattern recognize other pairs we know stand in the relation?
  - Does the new pattern return pairs that are not in the relation? (Precision)

Methods for relation extraction

1. Hand-written patterns
2. Machine Learning (Supervised classifiers)
3. Semi-supervised classifiers via bootstrapping
4. Semi-supervised classifiers via distant supervision
5. Unsupervised

4. Distant supervision for RE

- Combine:
  - A large external knowledge base, e.g. Wikipedia, Word-net
  - Large amounts of unlabeled text
- Extract tuples that stand in known relation from knowledge base:
  - Many tuples
- Follow the bootstrapping technique on the text

4. Distant supervision for RE

- Properties:
  - Large data sets allow for
    - fine-grained features
    - combinations of features

    $\text{M1} = \text{ORG}\ \&\ \text{M2} = \text{PER}\ \&\ \text{nextword=“said”}\&\ \text{path=}\ NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
  - Evaluation
- Requirement
  - Large knowledge-base

Methods for relation extraction

1. Hand-written patterns
2. Machine Learning (Supervised classifiers)
3. Semi-supervised classifiers via bootstrapping
4. Semi-supervised classifiers via distant supervision
5. Unsupervised

5. Unsupervised relation extraction

- Open IE
- Example:
  1. Tag and chunk
  2. Find all word sequences
     - satisfying certain syntactic constraints,
       - in particular containing a verb
     - These are taken to be the relations
  3. For each such, find the immediate non-vacuous NP to the left and to the right
  4. Assign a confidence score

United has a hub in Chicago, which is the headquarters of United Continental Holdings.

```
r1: <United,
has a hub in,
Chicago>
r2: <Chicago,
is the headquarters of,
United Continental Holdings>
```

Evaluating relation extraction

- Supervised methods can be evaluated on each of the examples in a test set.
- For the semi-supervised method:
  - we don’t have a test set.
  - we can evaluate the precision of the returned examples manually
- Beware the difference between
  - Determine for a sentence whether an entity pair in the sen-tence is in a particular relation
    - Recall and precision
  - Determine from a text:
    - We may use several occurrences of the pair in the text to draw a conclusion
    - Precision

We skip the confidence scoring

More fine grained IE

So far

- Tokenization+tagging
- Identifying the "actors"
  - Chunking
  - Named-entity recognition
  - Co-reference resolution
- Relation detection

Possible refinements

- Event detection
  - Co-reference resolution of events
- Temporal extraction
- Template filling

Some example systems

- Stanford core nlp: http://corenlp.run/
- SpaCy (Python): https://spacy.io/docs/api/
- OpenNLP (Java): https://opennlp.apache.org/docs/
- GATE (Java): https://gate.ac.uk/
  - https://cloud.gate.ac.uk/shopfront
- UDPipe: http://ufal.mff.cuni.cz/udpipe
  - Online demo: http://lindat.mff.cuni.cz/services/udpipe/
- Collection of tools for NER:
  - https://www.clarin.eu/resource-families/tools-named-entity-recognition

Today

- Information extraction:
  - Relation extractions
    - 5 ways
- Two words on syntax and treebanks
- Encoder-decoders
- Beam search

Sentences have inner structure

So far

- Sentence: a sequence of words
- Properties of words: morphology, tags, embeddings
- Probabilities of sequences
- Flat

But

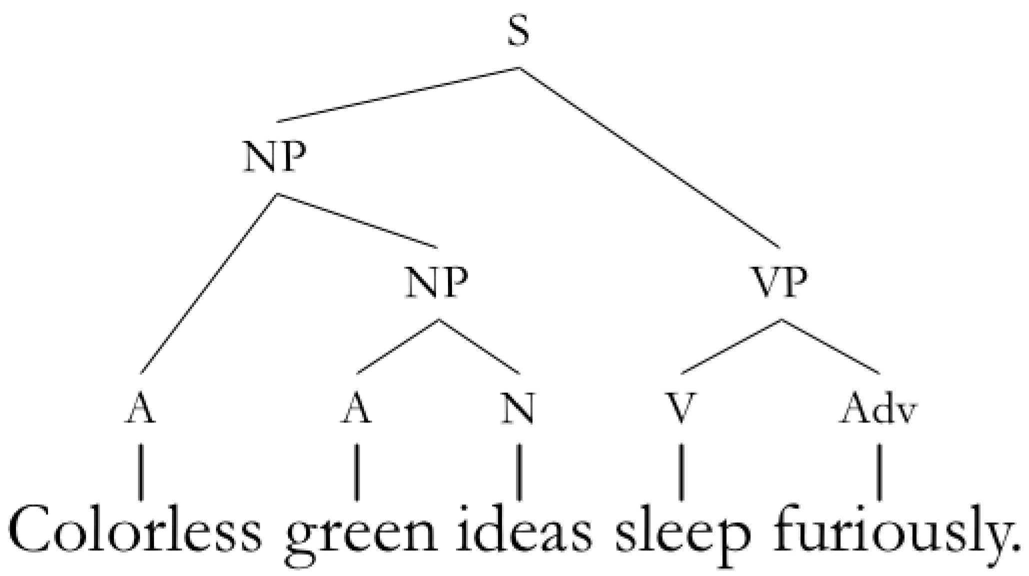
- Sentences have inner structure
- The structure determines whether the sentence is grammatical or not
- The structure determines how to understand the sentence

Why syntax?

- Some sequences of words are well-formed meaningful sentences.
- Others are not:
  - *Are meaningful of some sentences sequences well-formed words*
- It makes a difference:
  - *A dog bit the man.*
  - *The man bit a dog.*
- BOW-models don't capture this difference

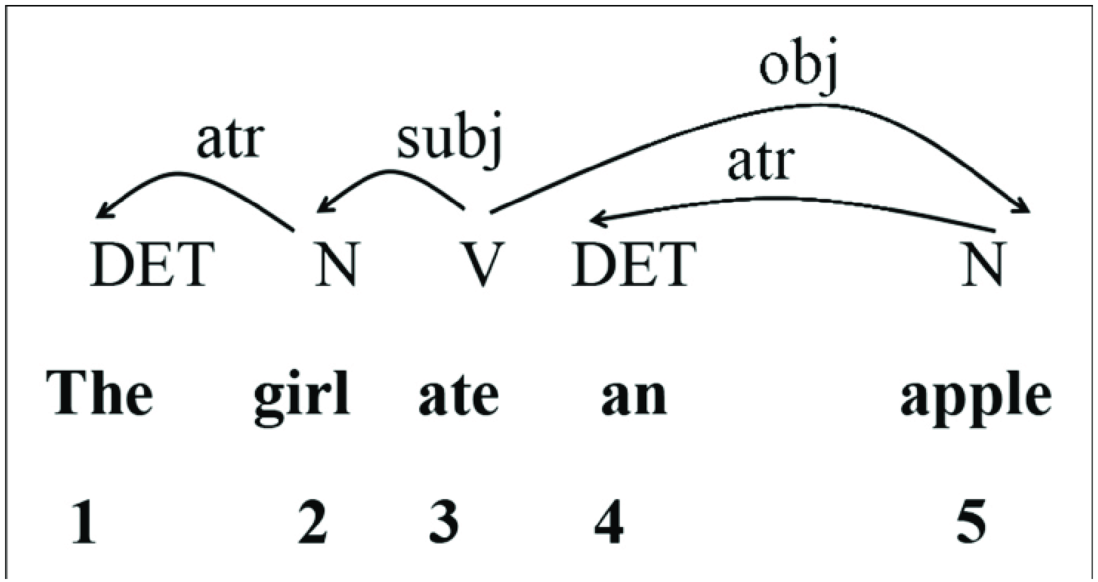
Two ways to describe sentence structure

Phrase structure



Focus of INF2820

Dependency structure



Focus of IN2110

Constituents and phrases

NP	VP	
	V	NP
Mary	*ate*	*the apple*
The small, cute dog	*saw*	*the small, cute dog*
The dog from Baskerville	*enjoied*	*the apple that Kim had stolen from the store*
You		*it*

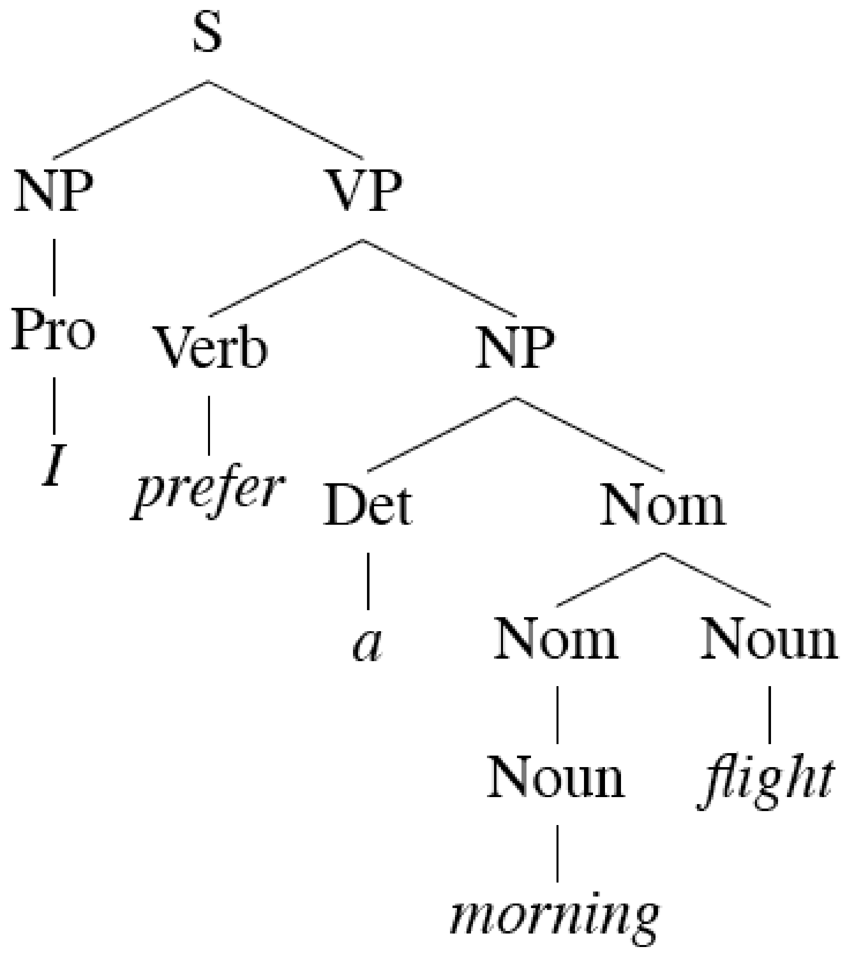
- Constituent: A group of word which functions as a unit in the sentence
  - See [Wikipedia: Constituent](#) for criteria of constituency
- Phrase: A sequence of words which "belong together"
  - = constituent (for us)
  - In some theories a phrase is a constituent of more than one word

Phrases

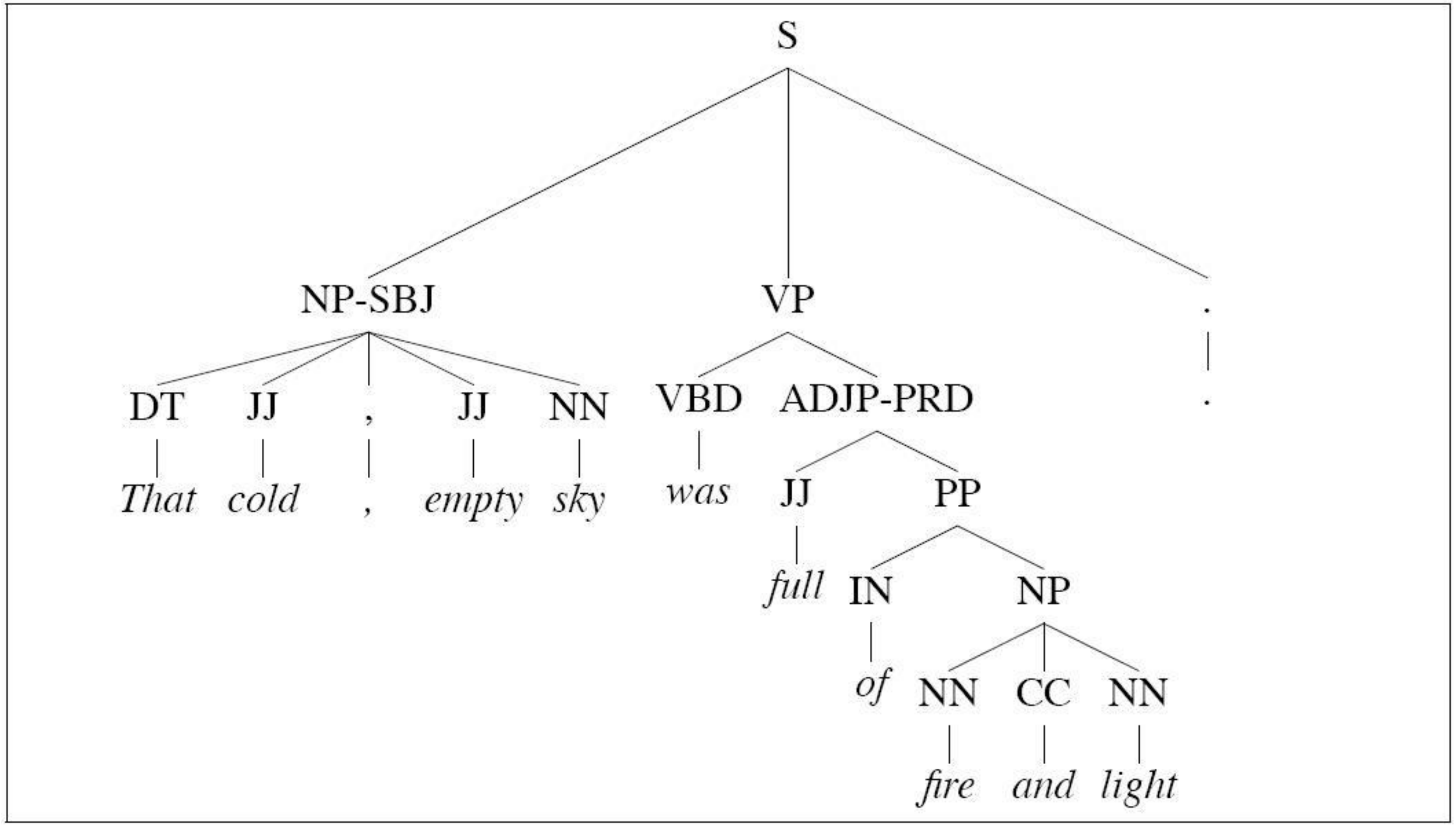
- Phrases can be classified into categories:
  - Noun Phrases, Verb Phrases, Prepositional Phrases, etc.
- Phrases of the same category have similar distribution,
  - e.g. NPs can replace names
  - (but there are restrictions on case, number, person, gender agreement, etc.)
- Phrases of the same category have similar structure, simplified:
  - NP (roughly): (DET) ADJ* N PP* (+ some alternatives, e.g. pronoun)
  - PP: PREP NP

Phrase structure

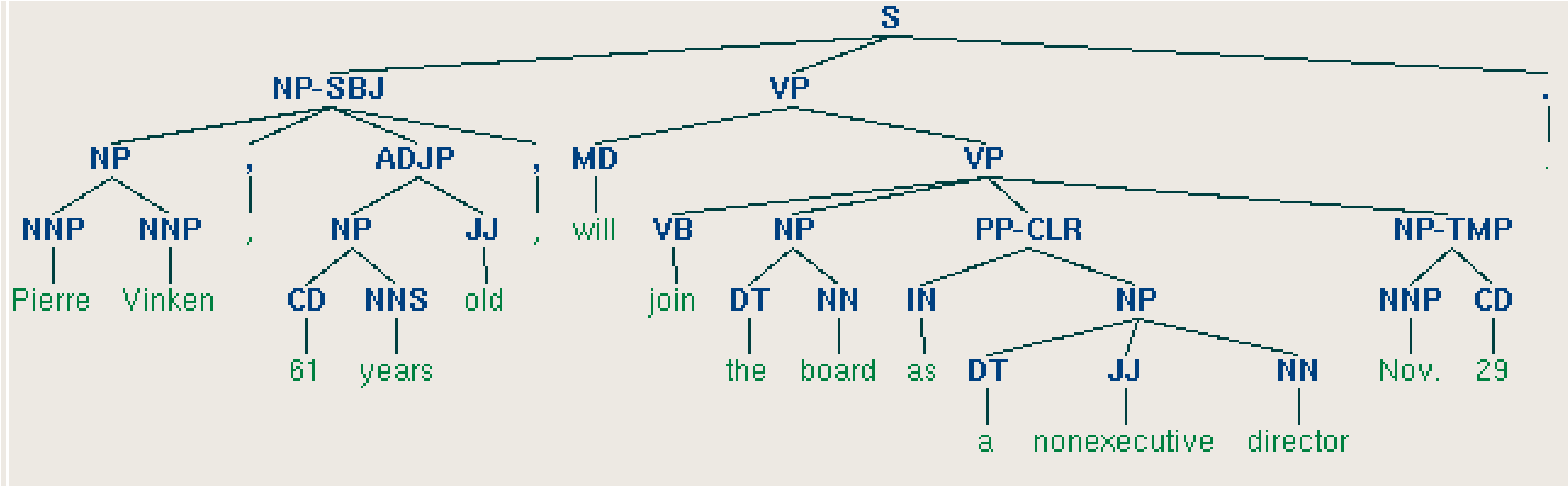
- A sentence is hierarchically ordered into phrases
- Various syntactic theories and models and NLP tools depart with respect to the actual trees:
  - Models based on X-bar theory prefer "deep threes": binary branching
  - Penn treebank prefers shallow trees



A Penn treebank tree



Treebanks



- A collection of analyzed sentences/trees
- Penn treebank is best known

Treebanks

- Treebanks are corpora in which each sentence has been paired with a parse tree (presumably the right one).
- These are generally created
  - By first parsing the collection with an automatic parser
  - And then having human annotators correct each parse as necessary.
- This requires detailed annotation guidelines that provide a POS tagset, a grammar and instructions for how to deal with particular grammatical constructions.

Different types of treebanks

Hand-made

- Human annotators assign trees.
- The trees define a grammar:
  - Many rules
  - Penn uses flat trees

Parse bank

- Start with a grammar
- And a parser
- Parse the sentences
- A human annotator selects the best analysis between the candidates
- May be used for training a parse ranker

Treebanks

- There are available free dependency treebanks for many languages
- The place to start in these days: http://universaldependencies.org/
- CONLL-formats:
  - One word per line, a number of columns for various information
  - CONLL-X, CONLL-U – different POSTAGs

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL
1	They	they	PRON	PRP	Case=Nom|Number=Plur	2	nsubj
2	buy	buy	VERB	VBP	Number=Plur|Person=3|Tense=Pres	0	root
3	and	and	CONJ	CC	_	4	cc
4	sell	sell	VERB	VBP	Number=Plur|Person=3|Tense=Pres	2	conj
5	books	book	NOUN	NNS	Number=Plur	2	obj
6	.	.	PUNCT	.	_	2	punct

from Andrei's INF5830 slides

Today

- Information extraction:
  - Relation extractions
    - 5 ways
- Two words on syntax and treebanks
- Encoder-decoders
- Beam search

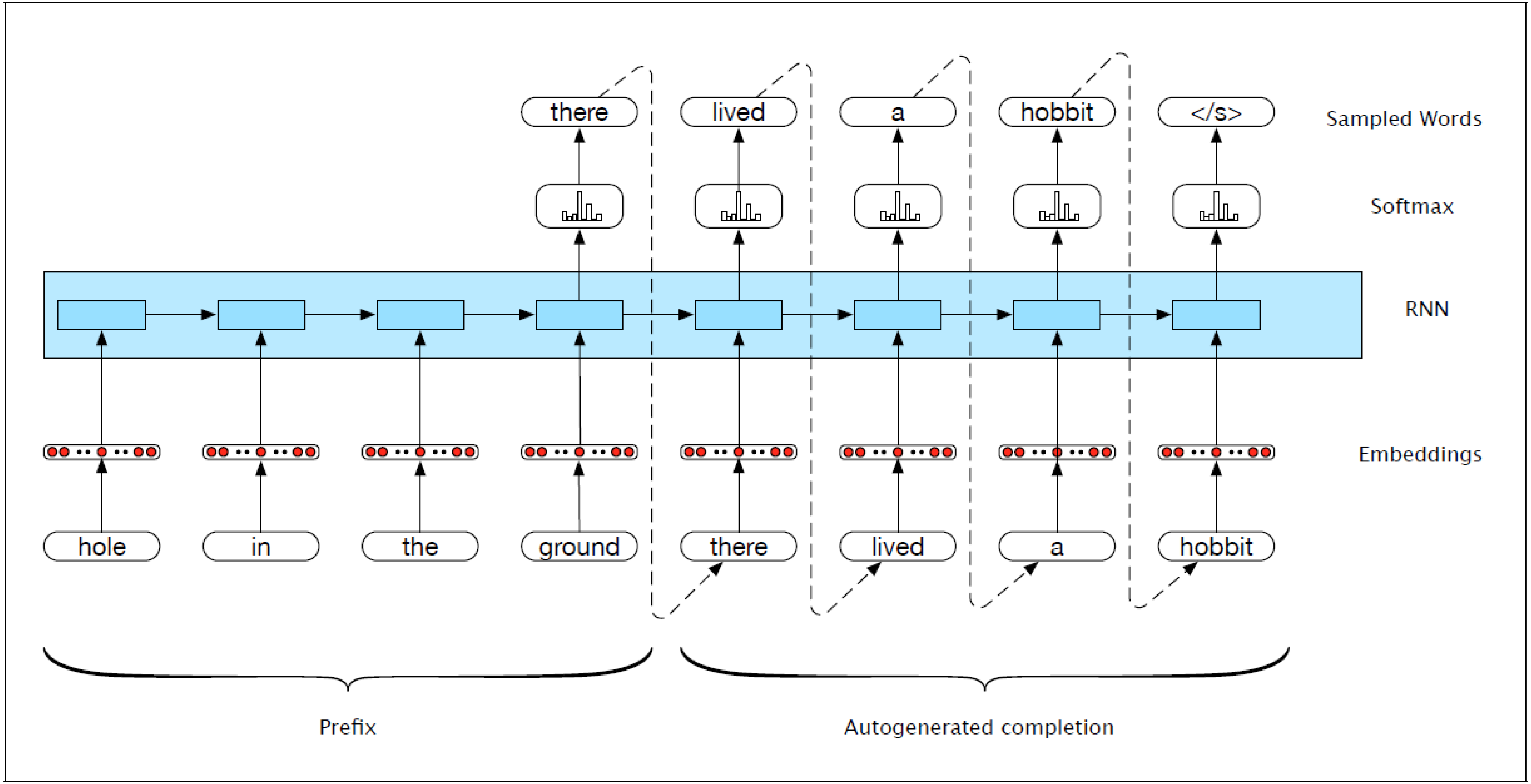


Figure 10.1 Using an RNN to generate the completion of an input phrase.

Idea

- Read-in the first part of the sentence, and
- then predict the rest of the sentence
- using an RNN trained on sentences

Applied to machine translation

- Bi-text
  - Text translated between two languages
  - The translated sentences are aligned into sentence pairs
- Machine learning based translation systems are trained on large amounts of bitext
- Encoder-decoder based translation
  - Concatenate the two sentences in a pair:
    - source sentence_<\s>_target sentence
  - Train an RNN on these concatenated pairs
  - Apply by reading a source sentences and from there predict a target sentence

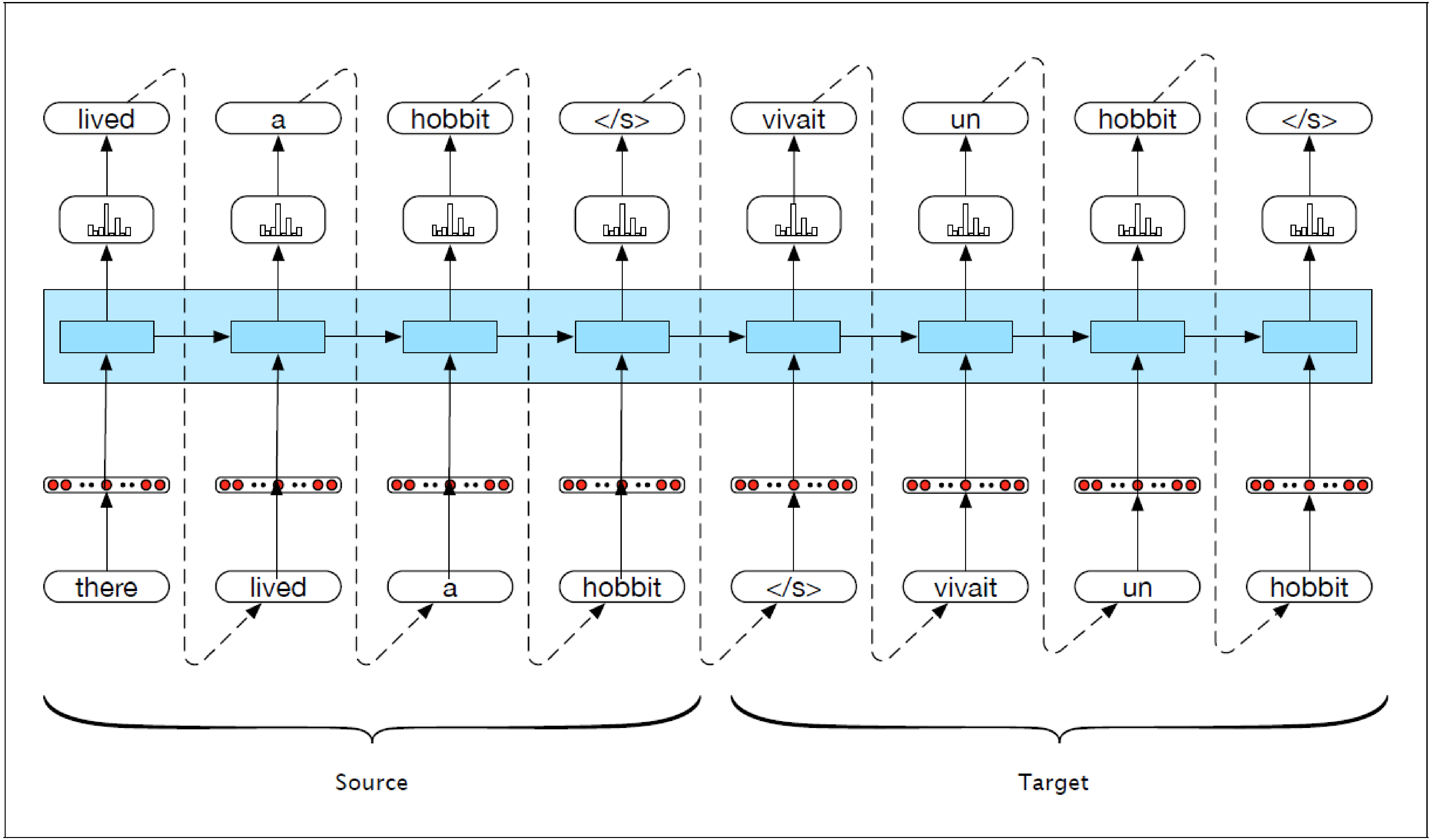


Figure 10.2 Training setup for a neural language model approach to machine translation. Source-target bitexts are concatenated and used to train a language model.

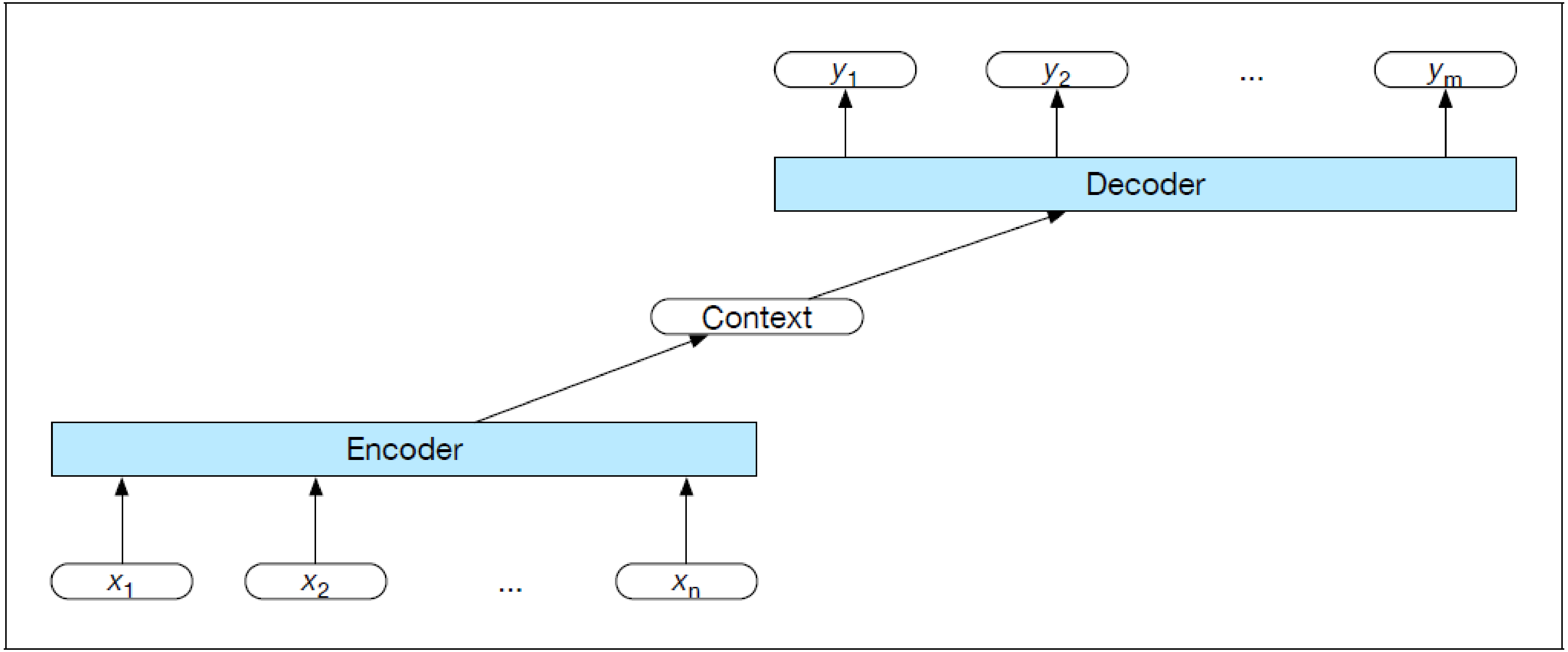


Figure 10.4 Basic architecture for an abstract encoder-decoder network. The context is a function of the vector of contextualized input representations and may be used by the decoder in a variety of ways.

Refinements

- The encoder can be more refined that a simple RNN,
  - e.g. bi-LSTM
  - (or using GRU which we will not consider here)
- The decoder may take more information into consideration

Today

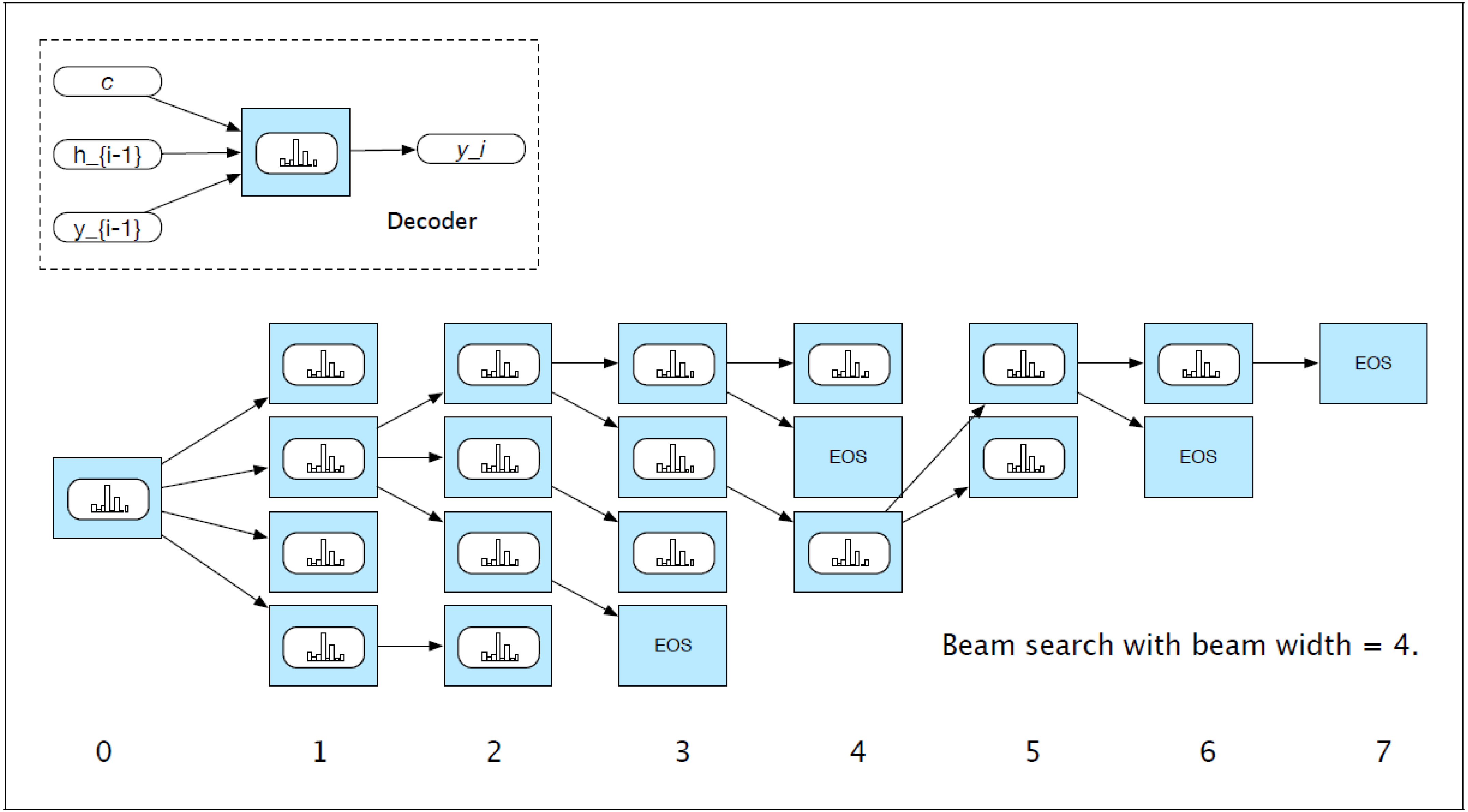
- Information extraction:
  - Relation extractions
    - 5 ways
- Two words on syntax and treebanks
- Encoder-decoders
- Beam search

Search

- For sequence labeling (tagging), we could use greedy search:
  - choose one label/tag at a time:
  - the most probable one given the ones we already have chosen
  - $\hat{t}_i = \underset{t_i}{\operatorname{argmax}} P\left(t_i \mid t_1^{i-1}, w_1^n\right)$
  - (the way we implemented the discriminative tagger in mandatory 2)
- But the goal is to find the most probable tag sequence given the data
  - $\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$
  - The HMM-model did this
  - If there is a limit to the history considered (e.g. *n* previous tags),
  - one can use a CRF-model for discriminative tagging, and dynamic programming as in HMM
- For encoder-decoder, there is no limit to the history, so this is not an option.

Beam Search

- Where greedy search chooses the unique best hypotesis at each step,
- Beam search keep a number of best hypotheses, say n=10
  - At each step it
    - considers the best continuations of these hypotheses
    - This will yield more than n hypotheses
    - it prunes away the less probable hypotheses, and keep the n best ones.


c
h_{i-1}
y_{i-1}
y_i
Decoder
EOS
EOS
EOS
EOS
EOS
Beam search with beam width = 4.
0
1
2
3
4
5
6
7