Natural Language Processing

Part of Speech

Mustafa Jarrar

Birzeit University

Watch this lecture and download the slides

Course Page: http://www.jarrar.info/courses/NLP/
More Online Courses at: http://www.jarrar.info

Acknowledgement:
Thanks to **Dima Taji** for helping me preparing this lecture

Natural Language Processing

Parts of Speech

In this lecture:

- Part 1: **POS Overview**
- Part 2: **POS Tagging (Viterbi algorithm)**
- Part 3: **ALMA Morph tagger**

What is Parts of Speech (POS)

Based on [1]

In traditional grammar, a part of speech (POS) is a category of words, or, more generally, of lexical items, that have similar grammatical properties.

Words that are assigned to the same part of speech generally display similar syntactic and semantic behavior, and sometimes they undergo morphological inflection similarly.

The History of Parts of Speech

The first description of parts of speech could be attributed to Dionysius Thrax of Alexandria (c. 100 B.C.), who described 8 parts of speech for Greek: **noun, verb, pronoun, preposition, adverb, conjunction, participle, and article**.

These POS tags became the basis for descriptions of European Languages.

Parts of Speech in Arabic

Traditional Arabic grammar includes three parts of speech:

noun (اسم), verb (فعل), Functional word (حرف/كلمة وظيفية).

Computationally, different annotation systems proposed different POS tagsets, inspired by the need to describe lexical units with varying granularities.

The decision of which POS tag set to select depends on the desired task.

Existing POS Tagsets in Arabic

1. Buckwalter tagset: contains around 500 unique tags (for stems and affixes), and untokenized (/combinations), this number gets inflated to over 300k unique tags.

Example :

(الجميلةَ): DET+ADJ+NSUFF_FEM_SG+CASE_DEF_ACC

This tagset is also used in the Penn Arabic Treebank.

There are several tagsets that evolved from the Buckwalter tagset with the goal of reducing the number of tagsets to a more manageable number, such as the **Bies** tagset, the **Kulick** tagset, and the **Extended Reduced** tagset.

Existing POS Tagsets in Arabic

1. Buckwalter tagset (Examples of Stem tags)

noun	اسم
noun_prop	اسم علم
adj	صفة
abbrev	إختصار
interj	تعجب
adv	ظرف
part	أداة
prep	حرف جر
conj	عطف
adj_comp	صفة مقارنة
noun_quant	اسم كم
noun_num	اسم عدد
adj_num	صفة عدد

IV	فعل مضارع
PV	فعل ماضي
CV	فعل أمر
IV_PASS	فعل مضارع مجهول
PV_PASS	فعل ماضي مجهول
VERB	فعل
INTERROG_PART+PV	أداة استفهام + فعل ماضي
PV+SUB_CONJ	فعل ماضي + أداة ربط
NEG_PART+PV	أداة نفي + فعل ماضي

Existing POS Tagsets in Arabic

2. Khoja Tagset:

177 tags: 103 nouns, 57 verbs, 9 particles (أدوات),
7 residuals (زوائد), and 1 punctuation.

Example:

باسمه (by his name) : PPr_NCSgMGI_NPrPSg3M

Existing POS Tagsets in Arabic

3. PADT tagset:

developed to for the **Prague Arabic Treebank**: 20 POS tags, and includes information on 7 features.

Example:

قالوا : VP-A-3MP

Existing POS Tagsets in Arabic

4. CATiB tagset:

developed for the Columbia Arabic Treebank, this tagset was designed with simplicity of annotation in mind, and therefore contains only 6 POS tags.

Example:

المدرسون: NOM

Comparison of Different Arabic POS Tagsets

Based on [2]

Arabic	Gloss	Buckwalter/PATB	CATiB	Bies	Kulick	ERTS	Khoja	PADT	ALMORGEANA/MADA
خمسون xams+uwna	fifty	NOUN_NUM+ NSUFF_MASC_PL_NOM	NOM	CD	CD	CD	NNuCaPlM	QL------1I	POS:NUM +MASC +PL +NOM
الف Âalf+a	thousand	NOUN_NUM+ CASE_DEF_ACC	NOM	CD	CD	CD	NNuCaSgM	QM-----S4R	POS:NUM +DEF +ACC
سائح sAŷiH+ī	tourist	NOUN+ CASE_INDEF_GEN	NOM	NN	NN	NNM	NCSgMGI	N------S2I	POS:N +INDEF +GEN
زاروا zAr+uwA	visited	PV+ PVSUFF_SUBJ:3MP	VRB	VBD	VBD	VBD	VPPl3M	VP-A-3MP--	POS:V +PV +S:3MS
مدينة madiyn+aħ+a	city	NOUN+ NSUFF_FEM_SG+ CASE_DEF_ACC	NOM	NN	NN	NNF	NCSgFAI	N------S4R	POS:N +FEM +SG +DEF +ACC
نا+ +nA	our	POSS_PRON_1P	NOM	PRP$	PRP$	PRP$	NPrPPl1	S----1-P2-	+P:1P
الجميلة Al+jamiyl+aħ+a	beautiful	DET+ADJ+ NSUFF_FEM_SG+ CASE_DEF_ACC	NOM	JJ	DT+JJ	DJJF	NASgFAD	A-----FS4D	POS:AJ Al+ +FEM +SG +DEF +ACC
في fiy	in	PREP	PRT	IN	IN	IN	PPr	P---------	POS:P
ايلول Âay.luwl+a	September	NOUN_PROP+ CASE_INDEF_GEN	PROP	NNP	NNP	NNPM	Rmy	N------S2I	POS:PN +INDEF +GEN
الماضي Al+mADiy	past	DET+ADJ	NOM	JJ	DT+JJ	DJJM	NASgMGD	A-----MS2D	POS:AJ Al+
. .	.	PUNC	PNX	PUNC	.	PUNC	PU	G---------	POS:PX

Existing POS Tagsets in Arabic

5. Qabas tagset:

Developed at SinaLab, based on SAMA, and used to tag Qabas and many corpora

See POS tagset and guidelines
https://sina.birzeit.edu/qabas/about

تصنيف قسم الكلام	أنواع المدخلات
اسمية	اسم
	اسم علم
	صفة
	صفة مقارنة
	صفة عدد
	اسم عدد
	اسم كم
	عدد
	صوت
	حرف-اختصار
	المجموع
فعلية	فعل ماضي
	فعل مضارع
	فعل أمر
	فعل ماضي مجهول
	فعل مضارع مجهول
	المجموع
وظيفية	ضمير ،ضمير إشارة،اسم موصول، ظرف موصول، ظرف،أداة استفهام،أداة استثناء،أداة تفصيل،أداة نداء، أداة تعريف، أداة ربط،أداة، حرف جر، حرف جر+ أداة تعريف، حرف جر+ أداة ربط،حرف جر+ أداة نفي، حرف جر+ اسم موصول، حرف جر+ضمير استفهام،حرف جر+ضمير إشارة،أداة ربط+ أداة نفي، عطف، أداة استقبال،أداة مضارعة، أداة نفي، أداة نفي+ اسم،فعل ماض+أداة ربط، شبه فعل+ اسم موصول، تعجب،جواب شرط+ أداة نفي، اسم فعل/ جامد، أداة فعل، شبه فعل،علامة ترقيم،أداة ربط+ أداة نفي.

Existing POS Tagsets in Arabic

5. Qabas tagset:

Nominals

اسم	NOUN
اسم علم	NOUN_PROP
صفة	ADJ
صفة مقارنة	ADJ_COMP
صفة عدد	ADJ_NUM
اسم عدد	NOUN_NUM
اسم كم	NOUN_QUANT
عدد	DIGIT
اسم صوت	NOUN_VOICE
حرف-اختصار	ABBREV

Verbs

فعل ماضي	PV
فعل مضارع	IV
فعل أمر	CV
فعل ماضي مجهول	PV_PASS
فعل مضارع مجهول	IV_PASS

Functional Words

ضمير	PRON
ضمير اشارة	DEM_PRON
اسم موصول	REL_PRON
ظرف موصول	REL_ADV
ظرف	ADV
اداة استفهام	INTERROG_PART
ظرف استفهام	INTERROG_ADV
ضمير استفهام	INTERROG_PRON
أداة إستثناء	RESTRIC_PART
أداة تفصيل	FOCUS_PART
أداة نداء	VOC_PART
أداة تعريف	DET
أداة	PART
أداة ربط	SUB_CONJ
حرف جر	PREP
حرف جر + أداة تعريف	PREP + DET
حرف جر + أداة ربط	PREP + SUB_CONJ
حرف جر + أداة نفي	PREP + NEG_PART
حرف جر + اسم موصول	PREP + REL_PRON
حرف جر + ضمير استفهام	PREP + INTERROG_PRON
حرف جر + ضمير اشارة	PREP + DEM_PRON
أداة ربط + أداة نفي	SUB_CONJ + NEG_PART
عطف	CONJ
أداة استقبال	FUT_PART
أداة مضارعة	PROG_PART
أداة نفي	NEG_PART
أداة نفي + اسم	NEG_PART + NOUN
فعل ماضي + أداة ربط	PV + SUB_CONJ
شبه فعل + اسم موصول	PSEUDO_VERB + REL_PRON
تعجب	INTERJ
ضمير تعجب	EXCLAM_PRON
جواب شرط + أداة نفي	RC_PART + NEG_PART
اسم فعل/جامد	VERB
أداة فعل	VERB_PART
شبه فعل	PSEUDO_VERB
علامة ترقيم	PUNC
إيموجي	EMOJI
جواب	-

Existing POS Tagsets in Arabic

5. Qabas tagset:

قواعد تصنيف المدخلات حسب قسم الكلام

أَولى معجم قبس أهمية كبيرة لتحديد قسم الكلام للمدخلات الواردة فيه، لما لذلك من أهمية كبيرة في التطبيقات الحاسوبية. ولم يكتفِ المعجم بتقسيم الكلمة على أنها اسم وفعل، بل توسع إلى أكثر من ذلك وخصص لكل منها وسوماً خاصة، فأدرج اسم العلم، واسم الفعل، والأداة، واسم العدد، والعدد، وغيرها من أقسام الكلام.فيما يلي شرح لكل تصنيف وقواعد ضبطه:

أولاً: الاسم

للاسم عشرة وسوم في معجم قبس، وفيما يلي وصف لهذه الوسوم:

اسم

41. يستعمل وسم ال(اسم) للمدخلات الاسمية التي ليست (اسم علم أو اسم عدد أو اسم كم أو صفة أو صفة مقارنة أو اسم صوت). يحتوي قبس على 36.29,053 مدخلة تم توسيمها باسم. مثل: شُعَيْبِيَّةٌ، مَأْمُونِيَّةٌ، تَسَطُّحٌ، مُرْتَعِبٌ، مُرْتَجَىً.

اسم علم

42. لا يستخدم اسم العلم كقسم كلام إلا في حالة واحدة فقط، وهي أن تكون المدخلة لا تحتمل إلا معنى اسم العلم. مثل: إِلْيَاسُ، سَمَاهِرُ، عِيسَىً، مُوسَى، زَكَرِيَّا، إِسْحَاقُ. يحتوي قبس على 4,319 اسم علم. أما إذا كانت المدخلة تحتمل عدة معانٍ وأحد هذه المعاني يفيد معنى اسم العلم فيجب إضافة تعريف يفيد اسم العلم. مثل: مدخلة مُصْطَفَى وبهذه الحالة لا يتم توسيمها باسم علم ولكن احدى معانيها هو اسم علم.

43. تضاف أسماء الأعلام المشهورة كأسماء المدن والدول والبحار والأنهار وغيرها كمدخلات، ويتم إضافة تعريف دلالي يعرف ويحدد هذه المدخلات/الأعلام - تم استخدام تعريفات من الويكيبيديا في حالة عدم توفر تعريفات معجمية جيدة.

اسم العدد

44. يستعمل للدلالة على الأعداد. يحتوي قبس على 44 اسم عدد، مثل: واحد، اثنان، ثلاث، سَبْعُونَ، سِتَّةٌ، عَشَرَةٌ، ثَمانِمَائَةٍ | ثَمَانِ مِأَئَةٍ.

صفة عدد

45. يستعمل للدلالة على الترتيب، مثل:حَادِي، أَوَّلٌ، عَاشِرٌ، سَابِعٌ، تَاسِعٌ. يحتوي معجم قبس على 12 مدخلة من نوع صفة عدد.

اسم كم

46. يستعمل للمدخلات التي تدل على الكمية، مثل: بِضْعٌ، بَعْضٌ، كُلّ، جَمِيعٌ. يحتوي معجم قبس على 19 اسم كم.

اسم صوت

47. تم إضافة أسماء الأصوات كمدخلات كما وردت في أغلب المعاجم، وإذا اختلفت بعض المعاجم في كتابتها، فيتم ترجيح المعاجم الأكثر شيوعاً، مثل: (حاي) لزجر الإبل،(بِس) للقطط، (بَس) لتسكين الإبل عند حلبها، (كخ) لزجر الطفل. يحتوي معجم قبس على 16 مدخلة تم توسيمها على أنها اسم صوت.

صفة

48. تستعمل إذا كانت جميع معاني الكلمة تشير إلى (صفة مشبهة، أو نسبة، أو اسم فاعل، أو اسم مفعول، أو تصغير، أو تكبير). ويتم الحكم على هذه الأنواع الستة بناء على الميزان الصرفي. ومثال ذلك: جَلِيٌّ، و بَارِدٌ ، و أَنِيقٌ التي تشير في جميع معانيها إلى معنى الصفة، لذا يتم توسيمها بصفة. أما مدخلة مُجَقَّعٌ فإن أحد معانيها ليس صفة، لذلك تم توسيمها بـ(اسم). يحتوي معجم قبس على 11,067 صفة. (انظر قاعدة رقم 9)

صفة مقارنة

49. يستعمل وسم صفة المقارنة مع المدخلات التي على وزن أَفْعَل التفضيل، مثل: أَعظَمُ. ويحتوي قبس على 295 مدخلة تم توسيمها بصفة مقارنة.

Existing POS Tagsets in Arabic

5. Qabas tagset:

الوسم	قاعدة الاستخدام	مثال
ضمير	يستخدم مع المدخلات الدالة على الشخص المتكلم أو المخاطب أو الغائب، إفرادا وتثنية وجمعا.	هو، هي، هما، هُم، هُنَّ، أنا، نَحْنُ، أنتَ، أنتِ، أنتما، أنتُم، أنتُنَّ، الياء، الهاء إيا من العامية: ولا، ولي، ولك، إحنا، إنتو.
ضمير إشارة	يستعمل مع المدخلات التي تشير إلى شخص أو شيء، مذكر أو مؤنث، قريب أو بعيد.	ولاء، تانِ، تانِك، تَيْنِ، تَيْنِك، تِيك، ذا، ذاك، ذَانِ، ذَانِك، ذَين، ذَيْنِك، ذِيك، ها، هاتِ، هَذا، هذه، هاتان، هذان، هؤلاء، أولئك، تلك، هنا، هناك، ثَمّ من العامية: هاد، هادي، هدول، هيدا، هيدي، هولي، هون، غاد هذاك، هديك، هذولاك
اسم موصول	اسم معرفة يستعمل مع المدخلات الدالة على شخص أو شيء محدد	الذي، التي، اللذان، اللتان، الذين، اللواتي، الألي، اللائي، ما، مَن، أيّ، ال، كم العامية: اللي، قديش
ظرف موصول	كلمة مركبة من ظرف واسم موصول، وتدل على شيء محدد يتم ذكره بعد الظرف الموصول، وتأتي بعده جملة (صلة الموصول) ولا يتم معنى العبارة إلا بذكر جملة صلة الموصول	حَيْثُما، كَم، كَيْفَ، كَيْفَما، كُلَّما، كُلَّما، إذما، لَم، مَتَى
ظرف	يستعمل مع المدخلات الدالة على زمن وقوع الحدث أو مكانه.	عند، بين، دون، حول، قبل، بعد، أمام، خلف، آن، أَبَدَ، آنذاك، أيضاً، إذاً، إذَنْ، إذّاك، بَعْدَ، ثُمَّ، ثَمَّةَ، حِينَئِذٍ، حِينَذاك، حِينَذاكَ، رُبَما، ساعَتَئِذٍ، سَنَةَ، سَوا، سِيَّما، طالَما، عامَئِذٍ، عامذاك، عَمَّ، عَنْدَ، عِنْدَئِذٍ، حيث، أين، متى، أنّى، أيان متى أين، مع قط فقط منذ لدى حوالي من العامية: قدام، هسا، هلا، هلقيت، بعدين، حوالين، لمّن، جوّا، بَس، بلكي أجنبية: أوت، أندر، وذ، أب.
أداة استفهام	يستعمل مع المدخلات المستخدمة للسؤال عن أمر غير معلوم.	أ، هَل، أنّى، أيان، أين، كَيْفَ، لِماذا، لِم، مَتَى، أيّ، كَم، ما، ماذا، مَنْ، ألا، إلام، يم، علام، عَمَّ، فيم، مِمَّ، ممّن، مَن، وبالعاميّة: إيش، أيمتى، شُحال، شْلون، شمعنا، ليش، ليه، مُو، مِين، وِين، وِيْنَنا، أنو، شُو، أني، شاكو، فش، ماكو، ها، وبلغات أخرى: لامّا، واي.
أداة استثناء	يستعمل مع الكلمات الدالة على إخراج شيء من حالة شيء آخر.	إلّا، إنَّما، سِوَى، غير، ليسَ، لَمَّا، عدا.
أداة تفصيل	يستعمل مع المدخلات الدالة على تفصيل الكلام.	أمّا
أداة نداء	يستعمل مع أدوات النداء.	أ، أيا، أيَّتُها، أيُّها، أيْ، يا، هَيا، وا، يا ، اللهمَّ، آيْ.

Natural Language Processing

Parts of Speech

In this lecture:

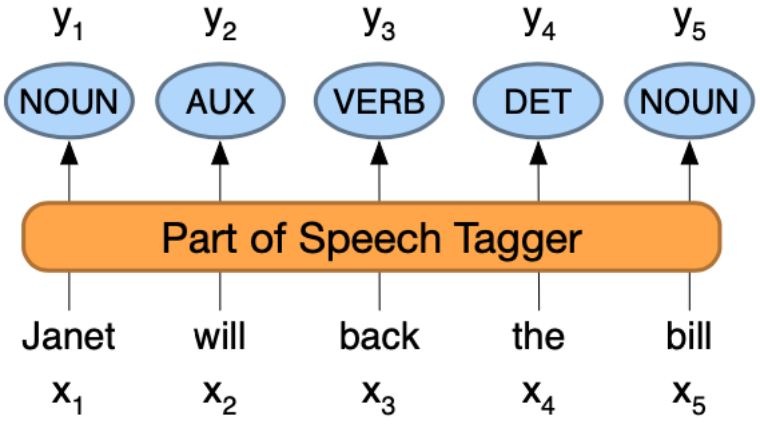
- Part 1: **POS Overview**
- Part 2: **POS Tagging (Viterbi algorithm)**
- Part 3: **ALMA Morph tagger**

Part-of-Speech Tagging

Based on [3]

Part-of-speech tagging is the process of assigning a part-of-speech to each word in a text.

The input is a sequence $x_1, x_2, \ldots, x_n$ of (tokenized) words and a tagset, and the output is a sequence $y_1, y_2, \ldots, y_n$ of tags, each output y_i corresponding exactly to one input x_i.



POS Tagging and Ambiguity

Given a word out of context, such as '**back**', POS tagging is a difficult process.

my **back** hurts? (as noun)

I will **back** you up'? (as verb)

Similarly, in Arabic, '**ذهب**' could have different tags depending on its contexts:

الولد إلى المدرسة **ذهب** (as verb)

دفعت ثمنه **ذهب** (as noun)

POS Tagging and Ambiguity

There are many methods for POS tagging:

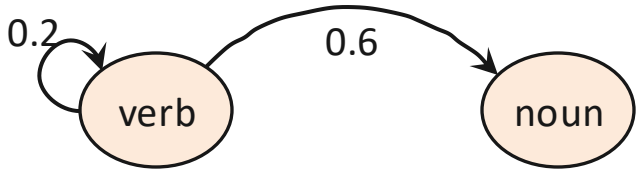
- The Hidden Markov Models using the **Viterbi algorithm** ← Classicial Method
- N-grams with SVM
- Frequency-based POS tagging ← ALMA Morph Tagger
- Sequence-to-Sequence using BERT (Transformers)
- Generative Models
- ….

POS Tagging Using Hidden Markov Model (HMM)

Based on [4]

HMM uses Markov Chains that operate on identifying a word's POS tag based on the POS tag of the previous word.

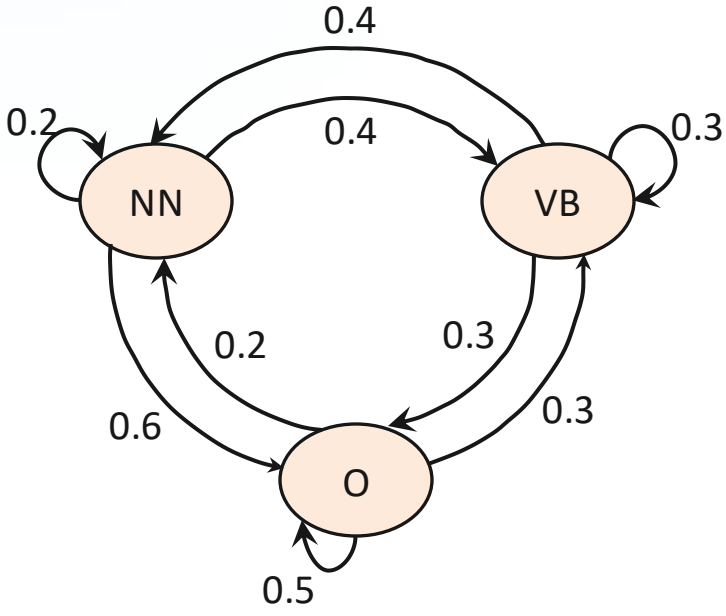
For example, in English, we know that a verb is more likely to be followed by a noun than another verb.



HMM Representation

Based on [4]

HMM are directed graphs, with weights on each edge:



	NN	VB	O
NN(noun)	0.2	0.2	0.6
VB(verb)	0.4	0.3	0.3
O(other)	0.2	0.3	0.5

Building a HMM

The *A matrix* contains the tag transition probabilities $P(t_i|t_{i-1})$ which represent the probability of a tag occurring given the previous tag, using maximum likelihood estimate:

$$\boldsymbol{P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}}$$

Example:

$$P(verb|noun) = \frac{C(noun\ verb)}{C(noun)}$$

These counts are extracted from large manually annotated corpora.

Building a HMM – cont.

The B matrix *contains the* emission probabilities $P(w_i|t_i)$, represent the probability, given a tag, that it will be associated with a given word.

In other words, what is the likelihood of a *verb* tag associating with the word 'back' versus a noun tag?

$$\boldsymbol{P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}}$$

Example:

$$P(back|verb) = \frac{C(verb\ back)}{C(verb)}$$

Using HMM

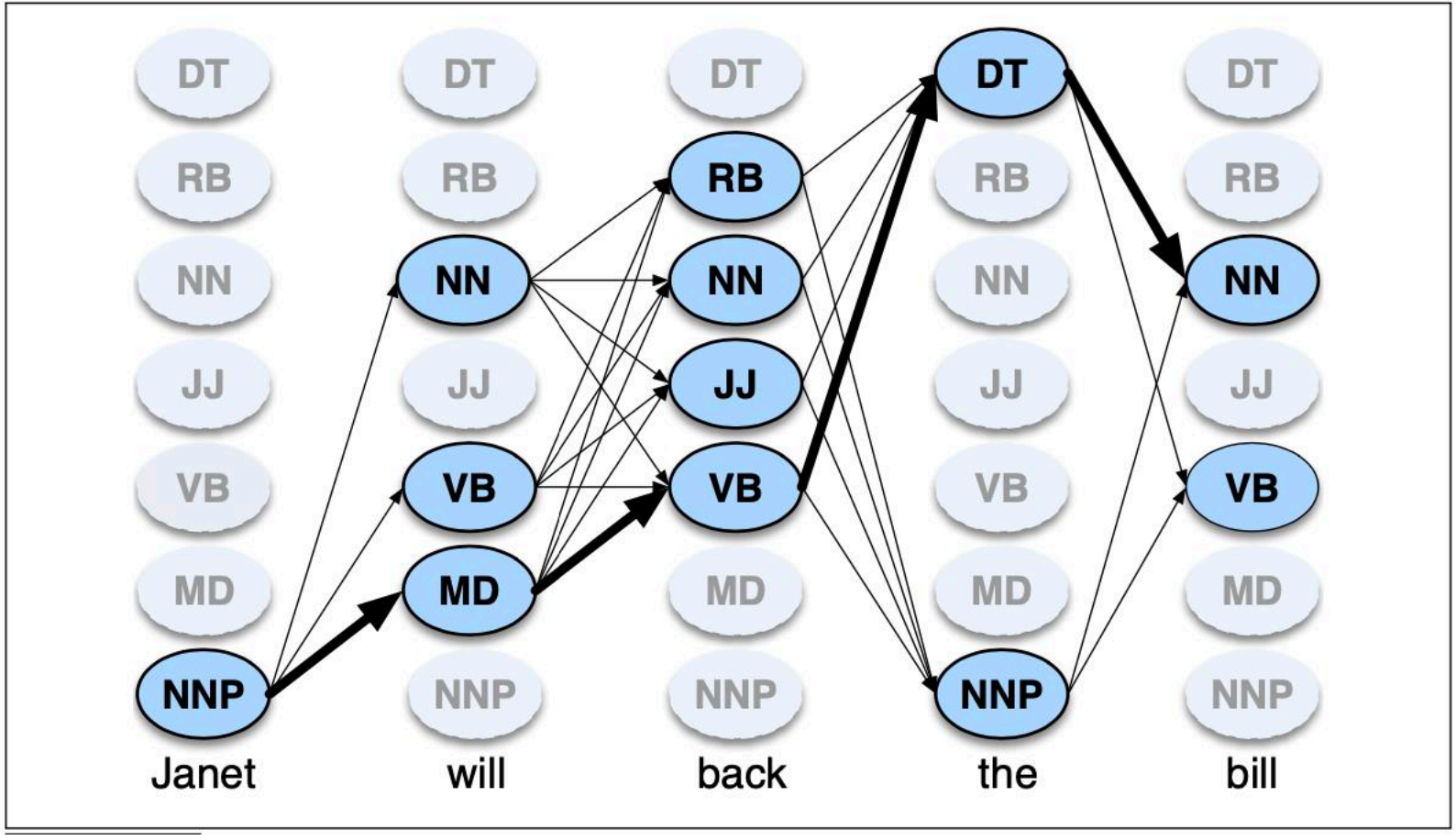
The simplest way to apply the HMM that is represented in matrices A (transition probability) and B (emission probability), is by using a greedy decoder called the **Viterbi algorithm**:

We pick that tag t_i that will maximize the value of:

$$P(t_i|t_{i-1}) * P(w_i|t_i)$$

POS Tagging using HMM with Viterbi Algorithm

Based on [3]



Viterbi Probability Computation

Based on [5]

Given a string *s* and token *t*

viterbi [s,t] =

$\max_{s'}$ (**veterbi**[*s'*,*t*-1] ×

transition probability P(*s* | *s'*) ×

emission probability P(token[*t*] | *s*)

)

For each s,t:

record which *s'*, t-1 contributed the max

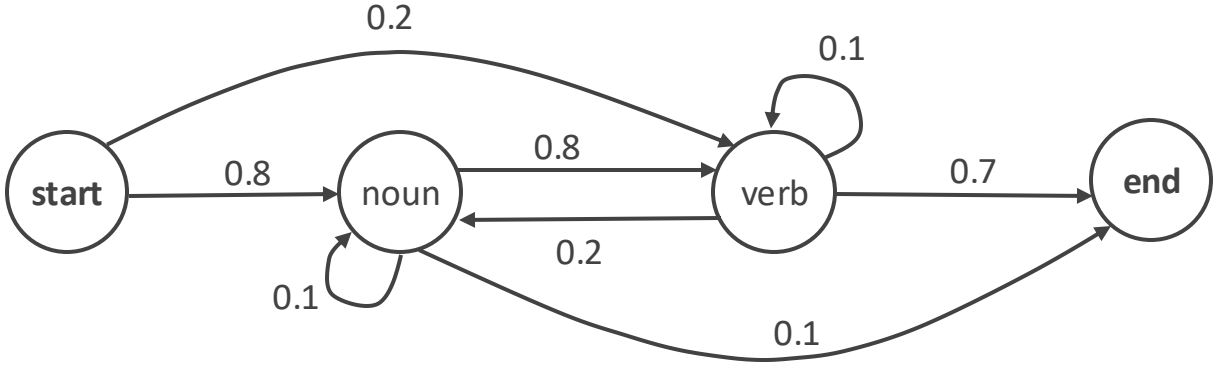
Analysis Example

Based on [5]

Tag the sentence:

Fish sleep.

Using the following simple HMM (transition probabilities):



Word Emission Probabilities

Based on [5]

A two-word language: "*fish*" and "*sleep*"

Suppose in our training corpus:

- "fish" appears 8 times as a noun and 5 times as a verb
- "sleep" appears 2 times as a noun and 5 times as a verb

Emission probabilities:

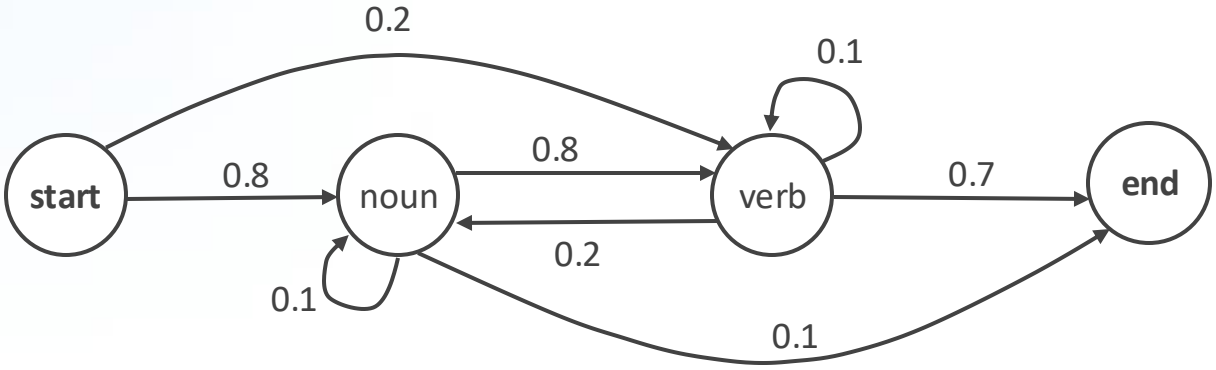
- Noun
  - P(fish | noun) : 0.8
  - P(sleep | noun) : 0.2
- Verb
  - P(fish | verb) : 0.5
  - P(sleep | verb) : 0.5

$$\frac{C(noun\ fish)}{C(noun)} = \frac{8}{2+8} = \frac{8}{10}$$

$$\frac{C(verb\ sleep)}{C(verb)} = \frac{5}{5+5} = \frac{5}{10}$$

Viterbi Probability

Based on [5]

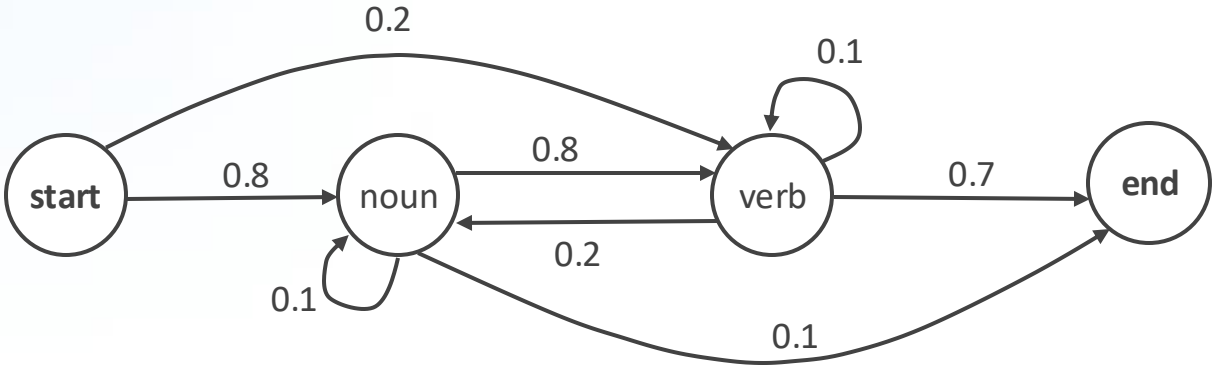


Fish sleep.

	0	1	2	3
start				
verb				
noun				
end				

Viterbi Probability

Based on [5]



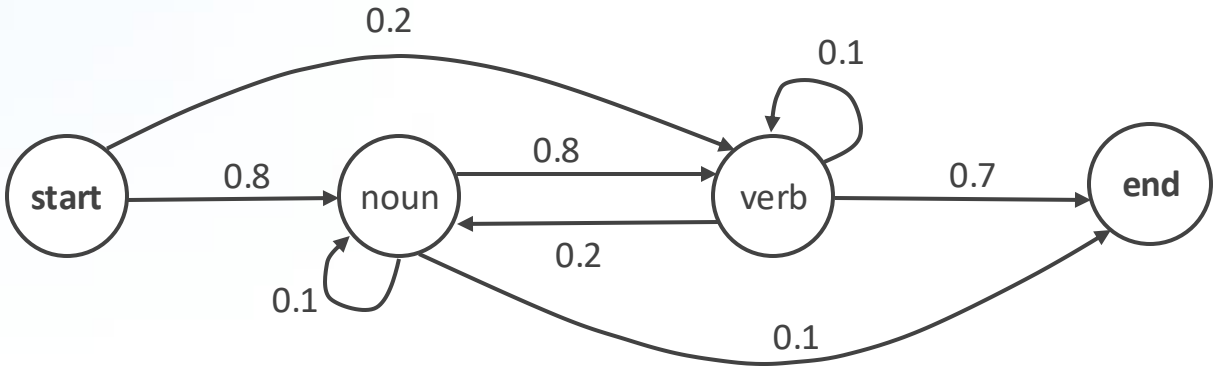
Fish sleep.

Current token: <start>

	0	1	2	3
start	1			
verb	0			
noun	0			
end	0			

Viterbi Probability

Based on [5]



Fish sleep.

Current token: fish

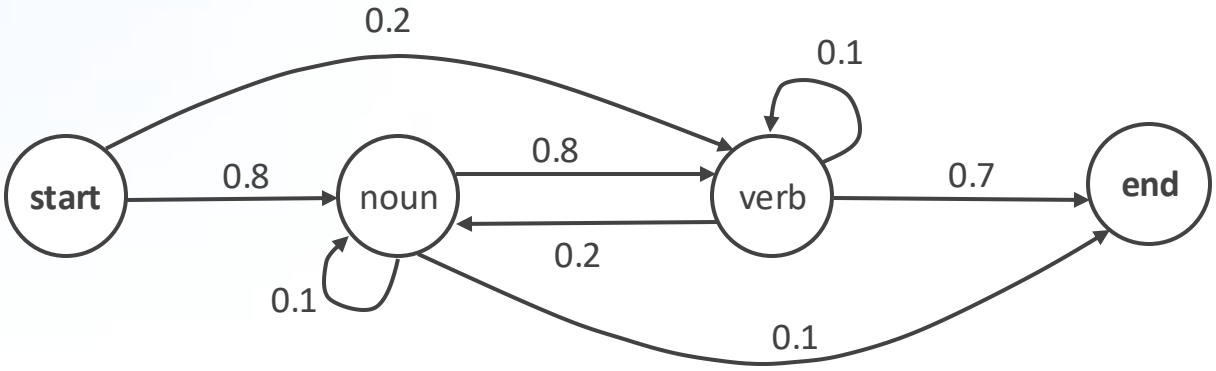
*transition * emission*

$$P(t_i|t_{i-1}) * P(w_i|t_i)$$

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1		
noun	0	0.8 * 0.8 = 0.64		
end	0			

Viterbi Probability

Based on [5]



Fish sleep.

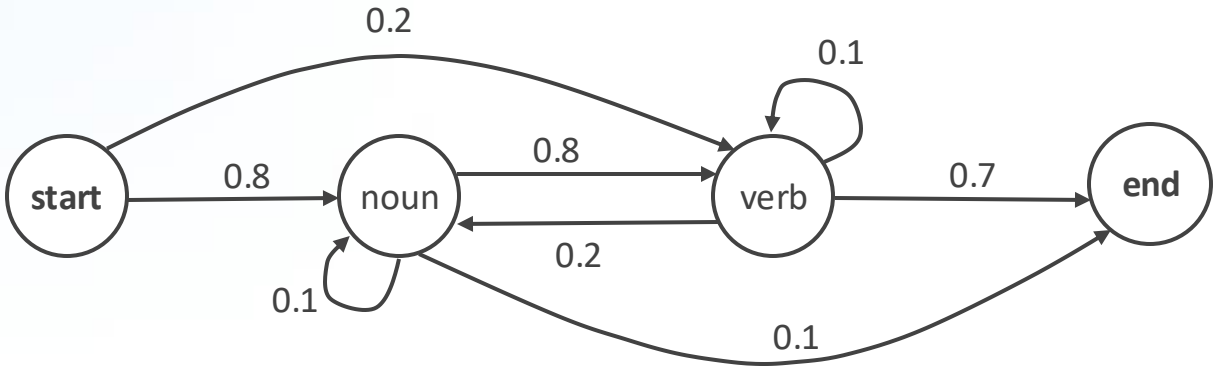
Current token: sleep

If fish was a verb

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	0.1 * 0.1 * 0.5 = 0.005	
noun	0	0.8 * 0.8 = 0.64	0.1 * 0.2 * 0.2 = 0.004	
end	0			

Viterbi Probability

Based on [5]



Fish sleep.

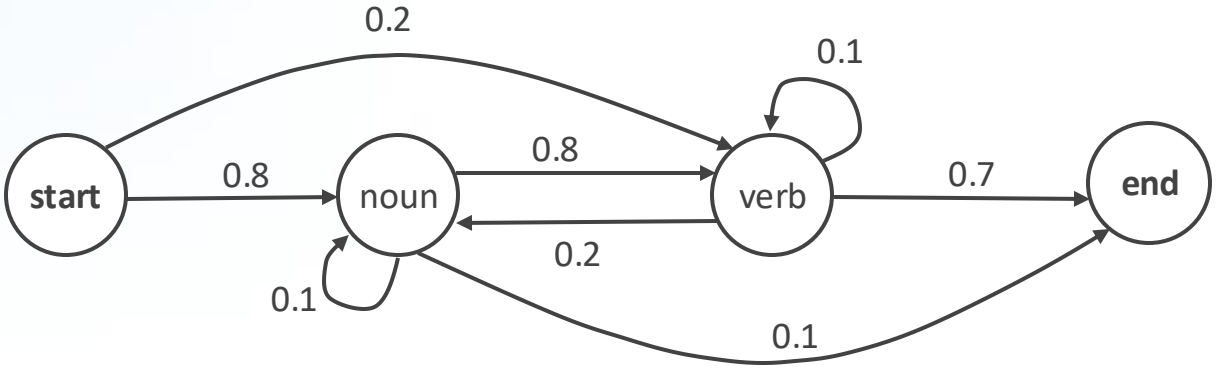
Current token: sleep

If fish was a noun

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	0.64 * 0.8 * 0.5 = 0.256	
noun	0	0.8 * 0.8 = 0.64	0.64 * 0.1 * 0.2 = 0.0128	
end	0			

Viterbi Probability

Based on [5]



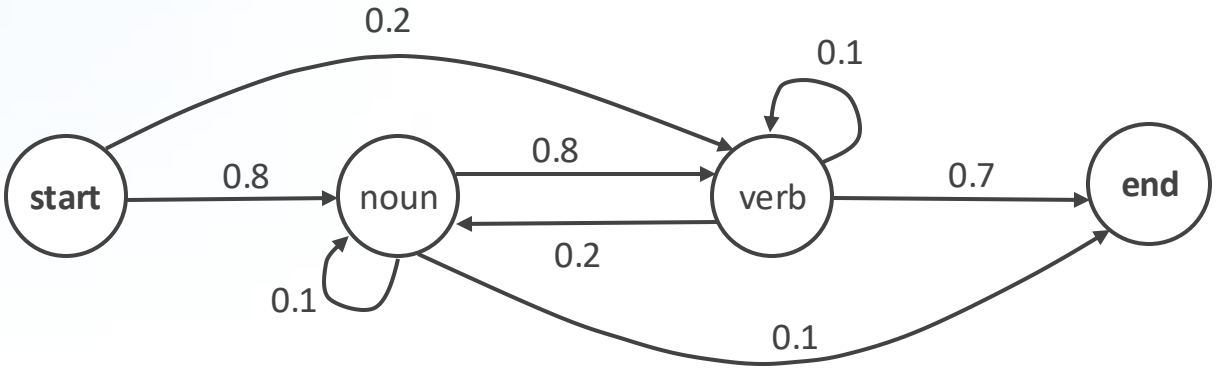
Fish sleep.

Current token: sleep

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			

Viterbi Probability

Based on [5]



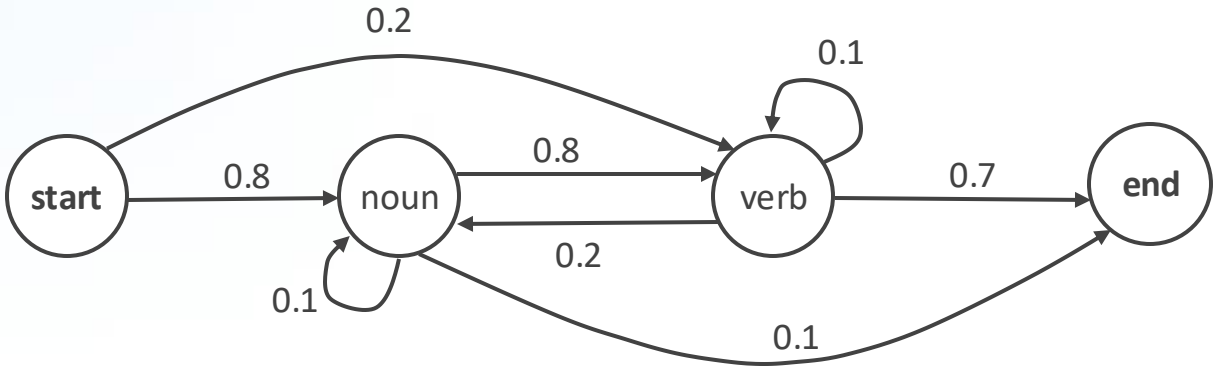
Fish sleep.

Current token: sleep

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			

Viterbi Probability

Based on [5]



Fish sleep.

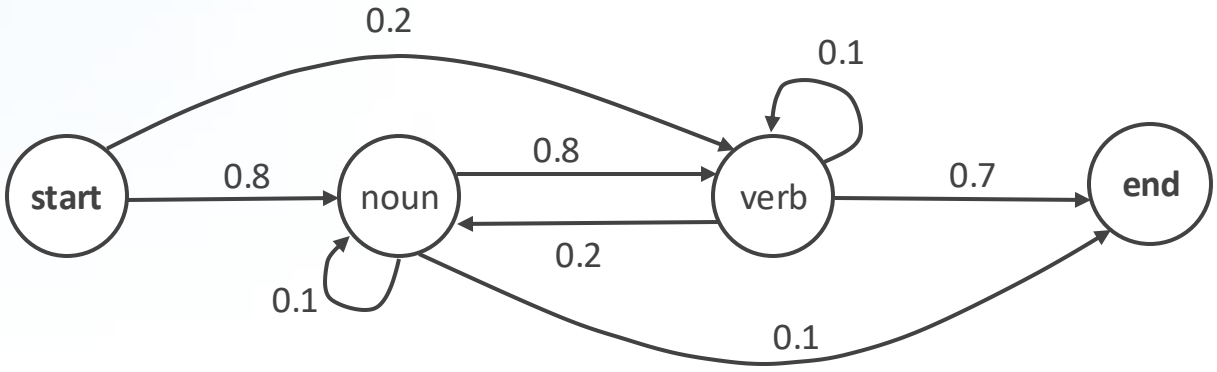
Current token: <end>

If sleep was a verb

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.256 * 0.7 = 0.1792

Viterbi Probability

Based on [5]



Fish sleep.

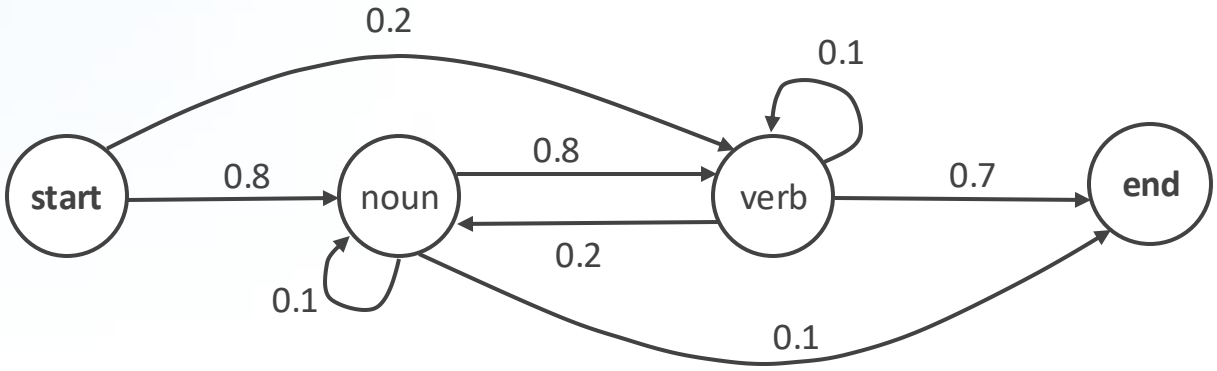
Current token: <end>

If sleep was a noun

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.0128 * 0.1 = 0.00128

Viterbi Probability

Based on [5]



Fish sleep.

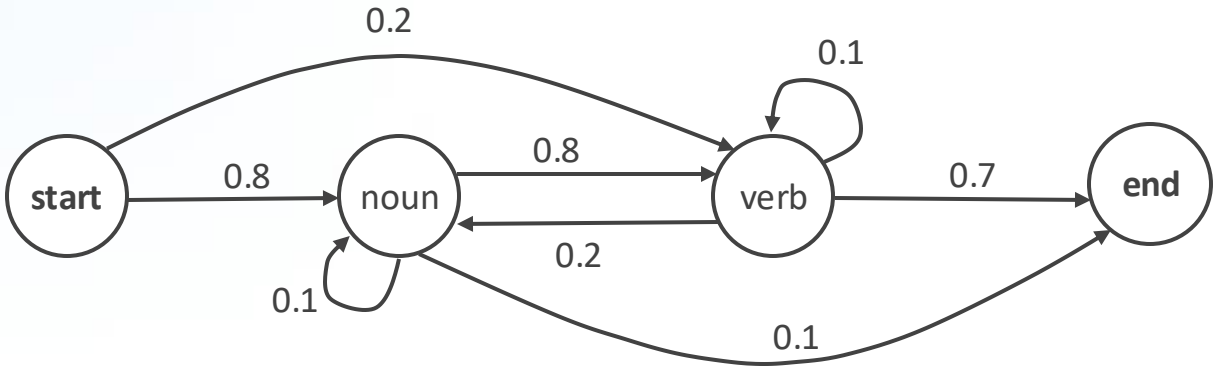
Current token: <end>

If sleep was a noun

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.1792 ~~0.00128~~

Viterbi Probability

Based on [5]



Fish sleep.

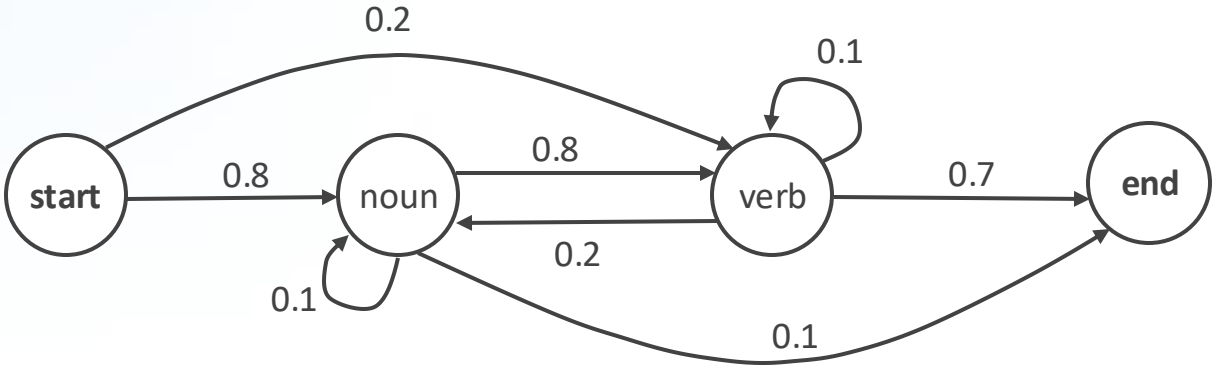
Current token: <end>

If sleep was a noun

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.1792 ~~0.00128~~

Viterbi Probability

Based on [5]



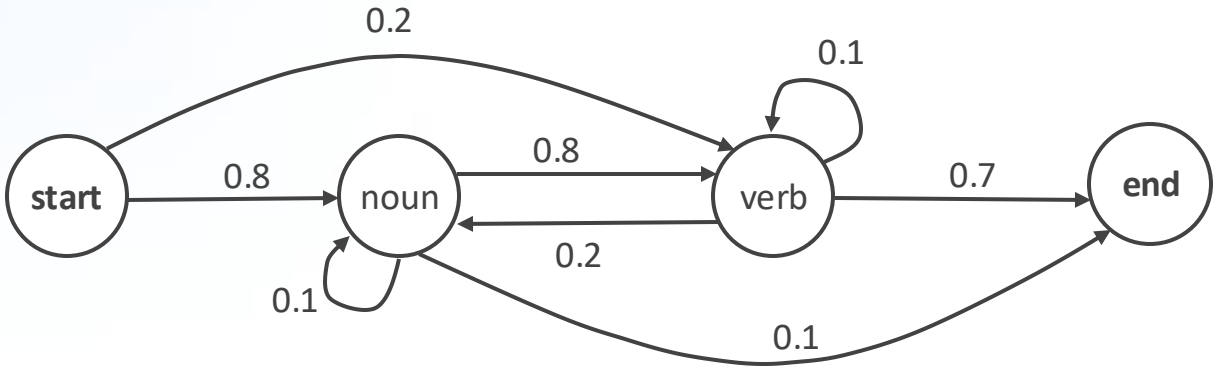
Fish sleep.

Decoding the result:
Sleep is a verb

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.1792 ~~0.00128~~

Viterbi Probability

Based on [5]



Fish sleep.

Decoding the result:
Sleep is a verb
Fish is a noun

	0	1	2	3
start	1			
verb	0	0.2 * 0.5 = 0.1	~~0.005~~ 0.256	
noun	0	0.8 * 0.8 = 0.64	~~0.004~~ 0.0128	
end	0			0.1792 ~~0.00128~~

Viterbi Complexity

The Viterbi algorithm complexity is $O(s^2n)$, where:

- s: number of states
- n: number of word/tokens in the input

Natural Language Processing

Parts of Speech

In this lecture:

- Part 1: **POS Overview**
- Part 2: **POS Tagging** (Viterbi algorithm)
- Part 3: **ALMA Morph Tagger**

ALMA Morphology Tagger

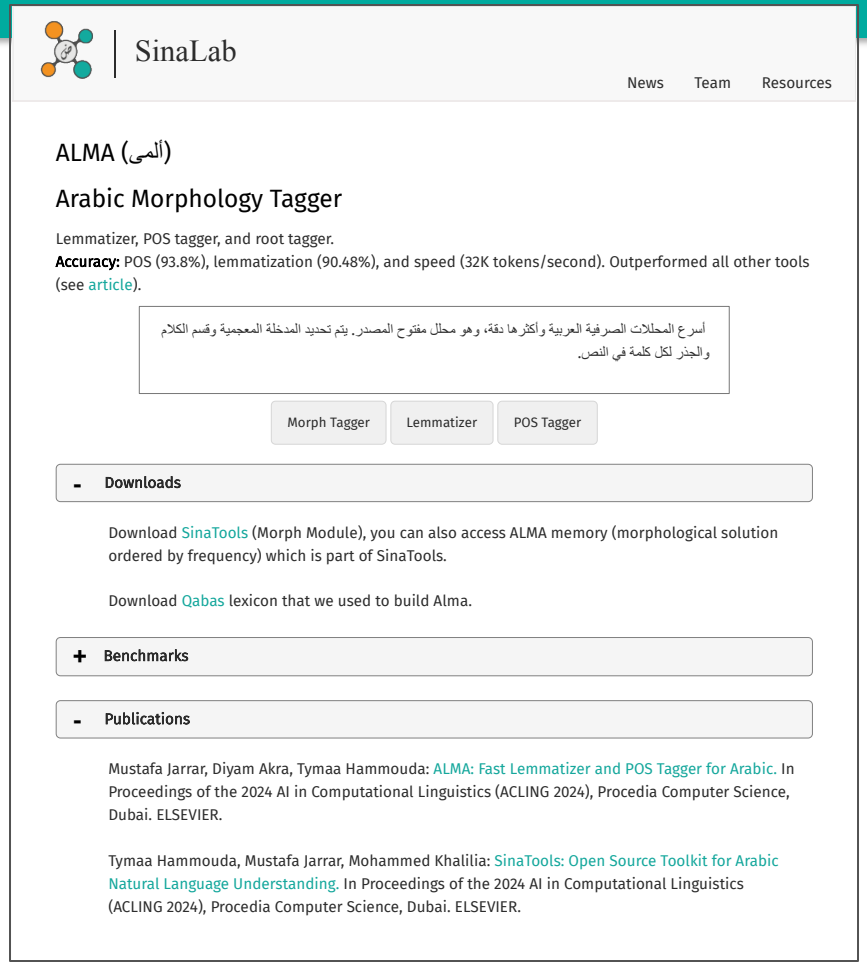
- **Tasks:** Lemmatizer, POS tagger, and root tagger (Frequency-based)
- **Open-Source**, part of SinaTools
- **Performance:** lemmatization (90.48%), POS (93.8%), speed (32K tokens/sec)
- **Benchmarking** the **accuracy** and **speed** of four tools
  - Alma, Madamira, CAMeL Tools, and Farasa
  - Using two corpora (ATB and Salma).

➔ **Alma outperformed all tools in all tasks.**

Mustafa Jarrar, Diyam Akra, Tymaa Hammouda: **ALMA: Fast Lemmatizer and POS Tagger for Arabic.** In Proceedings of the 2024 AI in Computational Linguistics (ACLING 2024), Procedia Computer Science, Dubai. ELSEVIER.

ALMA Online Demo

https://sina.birzeit.edu/alma/



Alma Memory

- Alma retrieves only the most frequent solution regardless of the context.
- All solutions for all Arabic words are collected and ordered by frequency.
- Alma's memory is a lookup (8.7M word forms, 10.7M solutions):


```
[ {"Word Form": "بيت",
   "Solutions": [{"lemma": " بَيْت3", "pos": "NOUN", "root": "ب ي ت", "frequency":596095},
                 {"lemma": " بَيْت1", "pos": "NOUN", "root": "ب ي ت", "frequency":140578},
                 {"lemma": " بَيْت4", "pos": "NOUN", "root": "ب ي ت", "frequency":9634},
                 {"lemma": " بَيَّت1", "pos": "PV", "root": "ب ي ت", "frequency":3153} ] },
  {"Word Form": "بلال",
   "Solutions":  [{"lemma":" بِلال1", "pos": "NOUN_PROP', "root": "ب ل ا ل",  "frequency": null} ]},
  {"Word Form": "احلام",
   "Solutions": [{"lemma":" حُلْم1", "pos": "NOUN", "root": "ح ل م", "frequency":212926},
                 {"lemma":" حِلْم1", "pos": "NOUN", "root": "ح ل م", "frequency":'304'} ] }]
```

ALMA System

Alma algorithm steps:

1. **Exact-match-search:** Search for exact match of a word.

2. **Cleaning-search-again-loop:** Sequentially applies following normalization steps: (1) removing (ال) prefixes, (2) replacing final (ة) with (ه), (3) unifying Alef variants, (4) removing diacritics and (5) shadda. Then search again.

3. **Out-of-vocabulary-lemmatization:** retrieves a word with ”_ 0” suffix.

4. **Out-of-vocabulary-root-tagging:** retrieves the wordform itself as root.

(Optional) Out-of-vocabulary-POS-tagging: predict a POS using a BERT model.

Alma

Part of SinaTools

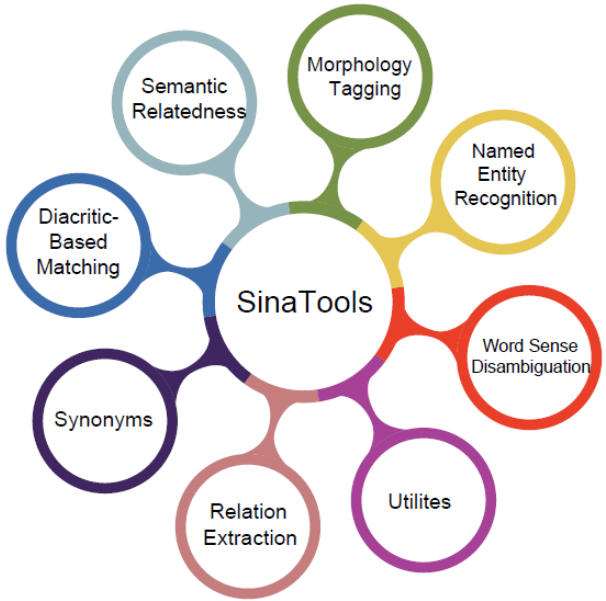
(Open Source)

SinaTools

Open-source Python toolkit for Arabic Natural Understanding, allowing people to integrate it into their system workflow.

Python APIs, command lines, and online demos.

Documentation | GitHub | Download | MIT License



Modules:

- + Morphology Tagging
- + Named Entity Recognition
- + Word Sense Disambiguation
- + Semantic Relatedness
- + Synonyms
- + Diacritic-Based Matching
- + Relation Extraction
- + Utilities

Morphology Tagging

SinaTools

Python API

```
from sinatools.morphology import morph_analyzer
morph_analyzer.analyze('ذهب الولد إلى المدرسة')
[{
    "token": "ذهب",
    "lemma": "ذَهَبَ",
    "lemma_id": "202001617",
    "root": "ذ ه ب",
    "pos": "فعل ماضي",
    "frequency": "82202"
  },{
    "token": "الولد",
    "lemma": "وَلَدٌ",
    "lemma_id": "202003092",
    "root": "و ل د",
    "pos": "اسم",
    "frequency": "19066"
  },{
    "token": "إلى",
    "lemma": "إِلَى",
    "lemma_id": "202000856",
    "root": "إ ل ى",
    "pos": "حرف جر",
    "frequency": "7367507"
  },{
    "token": "المدرسة",
    "lemma": "مَدْرَسَةٌ",
    "lemma_id": "202002620",
    "root": "د ر س",
    "pos": "اسم",
    "frequency": "145285"
}]
```

Benchmarking Datasets

We used two datasets:

LDC's Arabic Treebank (ATB): Part 3 V3.2

- **339,710** tokens (202,377 nouns, 32,135 verbs, 105,198 particles).
- annotated with features like lemma, POS, segmentation, and gloss.
- It has 13,031 unique lemmas (10,602 nouns, 2,229 verbs, 214 particles).
- News wires between 2001 and 2011.

Salma Corpus:

- **34,253** tokens(19,030 nouns, 2,763 verbs, 12,460 particles)
- annotated with various features including lemmas and POS.
- It has 3,875 unique lemmas (2,904 nouns, 677 verbs, 294 particles).
- MSA media sources between 2021 and 2023.

Lemmatization Benchmarking

Benchmarking challenges

- Both corpora use SAMA lemmas
- Four lemmatizers produce different lemma spellings
- MADAMIRA retrieves SAMA lemmas
- Alma lemmas are mapped into SAMA lemmas
- Lemmas in CAMeL Tools very close to SAMA lemmas without digits.
- Farasa drops digits, diacritics, Shadda, and Hamza

Lemmatization Benchmarking

Five scenarios of lemmatization: (بَيْت 1، بَيْت 3، بَيْت 4، بَيَّت 1) (قرأت بيتا من الشعر)

➔**True Lemmatization**: exact matching. (**بَيْت 3**)

➔**Ambiguous Lemmatization**: only digits are removing. (**بَيْت، بَيَّت**)

➔**Ambiguous undiacritized lemmatization**: all diacritics except Shadda are removing. (**بيت، بيّت**)

➔**Loose lemmatization:** Shadda is removing. (**بيت**)

➔**Baggy lemmatization:** all Hamza forms are normalizing.**(اناء) ➔(أناء، إناء)**

Lemmatization Benchmarking

Experiment	ATB	SALMA
Exp1: True Lemmatization (Exact Match)		
ALMA	**87.82%**	**90.48%**
MADAMIRA	84.18%	85.53%
CAMeL Tools	-	-
Farasa	-	-
Exp2: Ambiguous lemmatization (without Numbers)		
ALMA	**91.50%**	**91.17%**
MADAMIRA	88.35%	86.60%
CAMeL Tools	80.88%	81.73%
Farasa	-	-
Exp3: Ambiguous undiacritized lemmatization		
ALMA	93.25%	**93.40%**
MADAMIRA	**96.08%**	93.24%
CAMeL Tools	90.65%	90.83%
Farasa	68.92%	70.78%
Exp4: Loose Lemmatization (striped)		
ALMA	94.42%	**94.97%**
MADAMIRA	**96.73%**	94.48%
CAMeL Tools	93.64%	93.29%
Farasa	86.73%	91.28%
Exp5: Baggy lemmatization (striped and normalized)		
ALMA	95.01%	**95.88%**
MADAMIRA	**96.94%**	94.55%
CAMeL Tools	94.81%	94.73%
Farasa	94.86%	95.33%

Alma Lemmatization Errors

Types of Errors :

- *Ambiguous Lemmatization:* Alma retrieves (1 سِياسَة) while (2 سِياسَة) is the correct one.
- *Wrong Lemmatization:* Alma retrieves the lemma (هُمْ) for word (فهم).
- *Out of vocabulary Lemmatization:* word not found in Alma memory (ترامب)

Error Category	ATB Noun	ATB Verb	ATB Particle	ATB **All**	*Salma* Noun	*Salma* Verb	*Salma* Particle	*Salma* **All**
Ambiguous	39.2%[13501]	9.7%[3347]	11.9%[4077]	**60.7%[20925]**	36%[975]	7.0%[192]	18.0%[490]	**61.2%[1657]**
Wrong	23.8%[8208]	6.5%[2236]	8.9%[3063]	**39.2%[13495]**	20.6%[557]	11.7%[316]	6.4%[173]	**38.7%[1046]**
OOV	0.10 %[35]	0.009%[3]	0.02%[8]	**0.13%[46]**	0.18%[5]	0.04%[1]	-	**0.22%[6]**

POS Tagging Benchmarking

- Alma and MADAMIRA use the SAMA tagset (39 tags).
- CAMeL Tools uses these 39 tags but with different naming.
- Farasa uses a tagset of 18 tags.
- Three types of POS tagging scenario:
  - **True POS tagging:** used the full 39 tags used in the ATB
  - **Simplified POS tagging:** reduced the 39 tags into 13 simplified tags.
  - **POS Category tagging:** reduced the 39 tags into 3 tags/categories.

POS Tagging Benchmarking

	ATB		
	True POS	POS Simplified	POS Category
Alma	92.7%	93.1%	97.5%
Alma +BERT	**98.2%**	**98.6%**	**99.5%**
Alma +BERT(OOV)	92.7%	93.1%	97.6%
BERT	98.1%	98.6%	99.5%
MADAMIRA	82.3%	90.8%	97.5%
CAMeL Tools	82.7%	91.9%	98.6%
Farasa	62.4%	84.9%	94.3%

ALMA POS Tagging Errors

Types of Errors:

- General POS: (آليات) tagged as **ADJ** while its **NOUN**.
- Wrong POS: (أحب) tagged as **IV** while its **ADJ_COMP**.
- OOV: Alma fails to retrieve a POS for a word.

	ATB			
	Noun	Verb	Particle	All
General POS	$48.4\%^{10012}$	$4.5\%^{938}$	$6.2\%^{1281}$	$\mathbf{58.9\%}^{\mathbf{12172}}$
Wrong POS	$14.2\%^{2933}$	$14.9\%^{3075}$	$11.8\%^{2428}$	$\mathbf{40.9\%}^{\mathbf{8444}}$
OOV	$0.22\%^{47}$	$0.03\%^{8}$	$0.02\%^{5}$	$\mathbf{0.29\%}^{\mathbf{60}}$

Speed Benchmarking

- Speed of the four tools is evaluated:
  - **same machine** (24 CPU, 47G Memory, CentOS, size 1.3T)
  - **same setup** (reading input from a file and writing output to a file).
  - the **default settings** in all tools for lemmatization and POS tagging are used without changing any configuration.

- The experiment was repeated **six times**:
  - excluding the first run (initial model loading time is not considered).
  - the reading and writing of input/output files are considered

Speed Benchmarking

- Averages and standard deviations.
- Alma is significantly faster (34K tokens/second).
- This speed is attributed to ALMA's architecture, which uses a Python dictionary for fast memory access and processing.

Experiment	**ATB** (seconds)	**SALMA** (seconds)	**Speed** (tokens per second)
ALMA	$10^{\pm 1}$	$1^{\pm .11}$	33,997
MADAMIRA	$1710^{\pm 99}$	$370^{\pm 49}$	180
CAMeL Tools	$14,398^{\pm 1234}$	$677^{\pm 28}$	25
Farasa	$35^{\pm 2}$	$14^{\pm 2}$	7632

Natural Language Processing

Parts of Speech

In this lecture:

- ❑ Part 1: **POS Overview**
- ❑ Part 2: **POS Tagging** (Viterbi algorithm)
- ❑ Part 3: **ALMA Morph Tagger**
- ❑ Part 4: **Discussion**

References

[1] wikipedia: https://en.wikipedia.org/wiki/Part_of_speech

[2] Habash, N., : Introduction to Arabic NLP (2010)

[3] SLP, 3rd edition (Jurafsky and Martin, 2022)

[4] POS Tagging for Beginners, (Pykes, 2020)

[5] Based on NLP slides by Ralph Grishman, 2017

[6] Mustafa Jarrar, Tymaa Hammouda: Qabas: An Open-Source Arabic Lexicographic Database. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 13363–13370, Torino, Italia. ELRA and ICCL.

[7] Mustafa Jarrar, Diyam Akra, Tymaa Hammouda: **ALMA: Fast Lemmatizer and POS Tagger for Arabic.** In Proceedings of the 2024 AI in Computational Linguistics (ACLING 2024), Procedia Computer Science, Dubai. ELSEVIER.

[8] Tymaa Hammouda, Mustafa Jarrar, Mohammed Khalilia: **SinaTools: Open Source Toolkit for Arabic Natural Language Understanding.** In Proceedings of the 2024 AI in Computational Linguistics (ACLING 2024), Procedia Computer Science, Dubai. ELSEVIER.

[9] Jarrar, M. (2021). **The Arabic Ontology - An Arabic Wordnet with Ontologically Clean Content.** Applied Ontology Journal, 16:1, 1-26. IOS Press.

[10] Jarrar, M., & Amayreh, H. (2019). **An Arabic-Multilingual Database with a Lexicographic Search Engine**. In Proceedings – 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019). Lecture Notes in Computer Science (vol. 11608, pp. 234-246). Springer. Doi:10.1007/978-3-030-23281-8_19