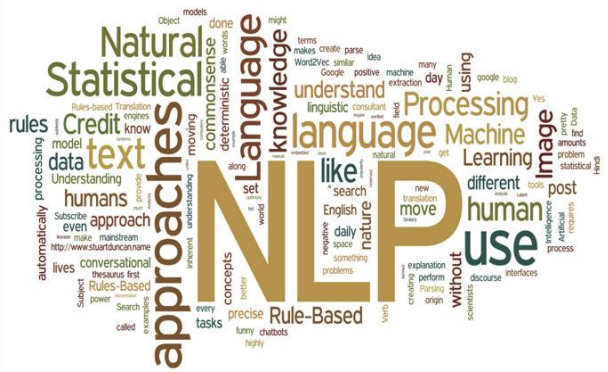


به نام خدا

گروه هوش مصنوعی، دانشکده مهندسی کامپیوتر

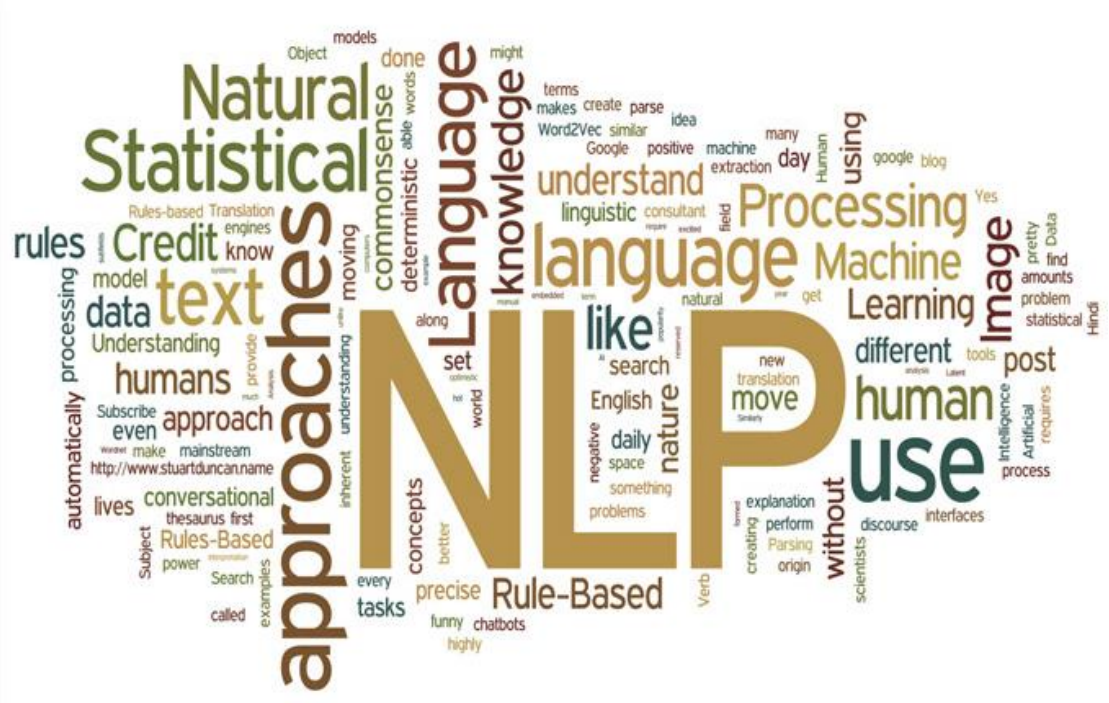
بخش دوم

پیش پردازش متن

حمیدرضا برادران کاشانی

اهداف این بخش:

- توکن‌بندی چیست و چگونه انجام می‌شود؟
- نرمالسازی کلمات و ریشه‌یابی چیست؟
- تقطیع جملات چگونه است؟



Hamidreza Baradaran Kashani

نرمالسازی متن (Text Normalization)

❖ در هر کاری مربوط به NLP، انجام ۳ گام زیر به عنوان نرمالسازی متن ضروری است:

1. تقطیع یا توکن‌بندی کلمات در متن (Segmenting/tokenizing words in running text)
2. نرمالسازی فرم یا شکل کلمات (Normalizing word formats)
3. تقطیع جملات در متن (Segmenting sentences in running text)

Hamidreza Baradaran Kashani

توکن بندی کلمات

(Word Tokenization)

Hamidreza Baradaran Kashani

متن چیست؟

- می‌توانیم متن را دنباله‌ای از:
  - کاراکترها (Characters)
  - کلمات (Words)
  - عبارات و موجودیت‌های نامدار (Phrases and name entities)
  - جملات (Sentences)
  - پاراگراف‌ها (Paragraphs)

- توکن بندی (Tokenization)
  - فرآیندی است که دنباله ورودی را به عناصر آن به نام توکن بخش‌بندی می‌کند.

Hamidreza Baradaran Kashani

کلمه چیست؟ مرز کلمات کجاست؟

- می‌توانیم کلمه را دنباله‌ای معنادار (meaningful) از کاراکترها در نظر گرفت.
- سوال مهم:

مرز کلمات چگونه پیدا می‌شود؟

- در زبان‌های مختلف مرز کلمات ممکن است با راهکارهای متفاوتی پیدا شوند.
- در انگلیسی و فارسی می‌توان از فاصله (space) یا علائم نقطه‌گذاری (punctuation) استفاده کرد. مثلا:

Input: Friends, Romans, Countrymen, lend me your ears;

Output: | Friends | Romans | Countrymen | lend | me | your | ears |

Hamidreza Baradaran Kashani

کلمه چیست؟ مرز کلمات کجاست؟

❖ در آلمانی با کلمات مرکب (**compound words**) مواجه هستیم که بدون فاصله نوشته می‌شوند:

“Lebensversicherungsgesellschaftsangestellter”

“life insurance company employee”

❖ **در زبان‌هایی مانند چینی و ژاپنی اصلا فاصله وجود ندارد.**

莎拉波娃现在居住在美国东南部的佛罗里达。

莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达

Sharapova now lives in US southeastern Florida

❖ جمله زیر به راحتی توسط انسان قابل خواندن است، اما برای کامپیوتر چه؟

Butyoucanstillreaditright

Hamidreza Baradaran Kashani

فرم‌های رایج کلمه؟

- دو فرم رایج برای تعریف کلمه می‌توان نشان داد:
  - **Word Type یا Type**
    - چه تعداد عناصر زبانی (عناصر موجود در دایره لغات یا **vocabulary** یک زبان) وجود دارند یا به بیان ساده چند کلمه یکتا وجود دارند؟
  - **Word Token یا Token**
    - چند نمونه از آن **Type** خاص وجود دارد؟

Hamidreza Baradaran Kashani

مثال

- چند کلمه در جمله زیر وجود دارد، از هر دو نوع **Token** و **Type**؟

They lay back on the San Francisco grass and looked at the stars and their

- Tokens: 15 or 14?
- Types: 13 or 12 or 11?

Hamidreza Baradaran Kashani

فرم‌های رایج کلمه؟

N = number of tokens

V = vocabulary = set of types

$|V|$ is the size of the vocabulary

Church and Gale (1990): $|V| > O(N^{½})$

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13 million

Hamidreza Baradaran Kashani

سایر چالش‌های توکن‌بندی

Finland's capital → Finland Finlands Finland's ?

what're, I'm, isn't → What are, I am, is not

Hewlett-Packard → Hewlett Packard ?

state-of-the-art → state of the art ?

Lowercase → lower-case lowercase lower case ?

San Francisco → one token or two?

m.p.h., PhD. → ??

Hamidreza Baradaran Kashani

الگوریتم حداکثر تطبیق (Maximum Matching)

- این الگوریتم بیشتر در توکن‌بندی در زبان چینی کاربرد دارد.

Given a wordlist of Chinese, and a string.

1) Start a pointer at the beginning of the string
2) Find the longest word in dictionary that matches the string starting at pointer
3) Move the pointer over the word in string
4) Go to 2

Hamidreza Baradaran Kashani

مثال حداکثر تطبیق

- Thecatinthehat
- Thetabledownthere

the cat in the hat

the table down there

theta bled own there

- برای زبان‌های انگلیسی و فارسی خوب کار نمی‌کند اما برای چینی خوب است.
- مثلا برای فارسی، نتیجه الگوریتم حداکثر تطبیق برای جمله زیر چیست؟

«دربهارباران‌داریم»

Hamidreza Baradaran Kashani

نکات ویژه در توکن بندی

Hamidreza Baradaran Kashani

علائم نقطه‌گذاری

آیا علائم نقطه گذاری را باید به عنوان توکن مجزا در نظر بگیریم یا آنها را حذف کنیم؟

✓ **پاسخ: بستگی به کاربرد و مساله مورد نظر دارد.**

❖ **در کاربرد آنالیز احساسات، وجود علائم نقطه گذاری به عنوان توکن مجزا می تواند مهم باشد.**

❖ **مثلا دو جمله زیر با در نظر گرفتن علامت نقطه یا علامت سوال معنا و حس متفاوتی را ایجاد می کنند.**

"I hate dogs."

"I hate dogs?"

❖ **پس در اینجا هر دو جمله باید به ۴ توکن تبدیل شوند.**

"I hate dogs?" → ["I", "hate", "dogs", "?"]

توکن بندی

❖ **در عمده ماژول های پیاده سازی شده مثلا در CountVectorizer تولباکس Sklearn ، بصورت پیش فرض علائم نقطه گذاری در فرآیند توکن بندی حذف می شوند.**

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر فاصله

اگر فقط از فاصله (whitespace) برای توکن بندی استفاده کنیم چه مشکلاتی ممکن است بوجود آید؟

["I", "hate", "dogs?"]

["I", "hate", "dogs."]

- در واقع باید دو تا توکن مجزا در نظر بگیریم یکی برای "dogs." و دیگری برای "dogs?".
- در نتیجه تعداد کلمات در واژگان ما زیاد می شود.
- پس بعد بردار ویژگی افزایش می یابد.
- در نتیجه نیاز به داده بیشتری برای یادگیری مدل یادگیری ماشینی (که بر روی توکن ها اعمال می شود) خواهیم داشت.
- به عبارتی باید تعداد زیادی جمله داشته باشیم که کلمه **dogs** با نقطه پایانی دارد و تعداد زیادی جمله که **dogs** با علامت سوال است.

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر کاراکتر

- تا اینجا عمدتا فرض شده است که توکن ها کلمات هستند. البته این لزوما بهترین نیست!
- در ادامه به برخی از مزایا و معایب توکن بندی مبتنی بر کلمات در مقابل توکن بندی مبتنی بر کاراکتر اشاره می کنیم.

- در توکن بندی مبتنی بر کلمه نیاز به ساخت و ذخیره سازی واژگان بزرگ با اندازه 1M کلمه داریم !
- تعداد 1M بردار تعبیه کلمه با بعد بسیار زیاد 1M !!
- برای ساخت یک مدل زبانی با شبکه عصبی عمیق ممکن است نیاز باشد تا شبکه عصبی در خروجی خود یک توزیع احتمال روی تمام کلمات درون واژگان را بدهد. یعنی یک توزیع احتمال با 1M مقدار!!!
- این مساله معادل با یک ماتریس وزن با ابعاد خیلی بزرگ در انتهای شبکه است (خوب نیست!!)

ایراد توکن بندی مبتنی بر کلمات

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر کاراکتر

- کلمات معنا دارند و حاوی اطلاعات زیادی هستند.
- برای مثال وقتی کلمه "کبوتر" را می بینیم، می توانیم خیلی سریع و بدون ابهام در ذهن مان تصویرسازی کنیم.

مزیت توکن بندی مبتنی بر کلمات

- در مقابل کاراکترها حاوی اطلاعات زیادی نیستند.
- فرضا وقتی کاراکتر **d** را می بینیم می دانیم این کاراکتر در کلمات مثل **dog**، **duck** و **door** به کار رفته است. در حالیکه این کلمات کاملا به مفاهیم متفاوتی اشاره دارند. بنابراین در نهایت این توکن به تنهایی در ارتباط با مفهوم یا معنای خاصی نیست.

ایراد توکن بندی مبتنی بر کاراکتر

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر کاراکتر

- تعداد توکن های کاراکتری بسیار کم است.
- مثلا در انگلیسی ۲۶ تا حرف داریم به علاوه تعداد محدودی کاراکتر برای فاصله و علائم نقطه گذاری
- در نتیجه واژگان ساخته شده بر اساس کاراکترها بسیار کوچک است (به مراتب کمتر از یک میلون توکن کلمه).
- لذا امکان ذخیره سازی و پردازش آنها توسط کامپیوترها براحتی وجود دارد.

مزیت توکن بندی مبتنی بر کاراکتر

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر زیرکلمه

- زیرکلمه ها واحدهای میانی هستند بین توکن های کاراکتری و توکن های کلمه ای
- زیرکلمه ها با تقسیم کلمه ها به واحدهای کوچکتر سازنده آنها حاصل می شوند، مثلا:

"eating" → "eat" + "ing"

- دو کلمه **"eat"** و **"eating"** بسیار به هم نزدیک و مرتبط هستند، بنابراین ما معمولا می خواهیم آنها بازنمایی کلمه یکسان و مشترکی برای الگوریتم یادگیری ماشین مان داشته باشند.
- اگر **"eating"** را به واحدهای زیرکلمه ای توکنایز نکنیم، چه مشکلی ممکن است پیش آید؟
- در اینصورت دو کلمه **"eat"** و **"eating"** بصورت دو کلمه کاملا متفاوت بازنمایی می شوند و دیگر لزوما شباهت بردارهای بازنمایی این دو کلمه بیشتر نیست از شباهت میان کلمه **"eat"** با کلمه نامرتبط دیگری مثل **"table"**

Hamidreza Baradaran Kashani

توکن بندی مبتنی بر زیرکلمه

❖ به بیان دیگر ما فقط باید امیدوار باشیم که مدل یادگیری ماشین با جملات مختلفی که به آن می دهیم تشخیص دهد که این دو کلمه باید مشابه باشند (لزوما ممکن است این اتفاق نیفتد!)

❖ سوال؟

❖ آیا ما می خواهیم مدل یادگیری ماشین ما کلماتی مثل **walk**، **walking**، **walked** و **walks** را بصورت جداگانه یاد بگیرد؟ یا می خواهیم از طریق یک بازنمایی مشترک یا مشابه به همدیگر مرتبط شوند؟

❖ برای پاسخ به این سوال باید منتظر بمانیم تا در بحث یادگیری عمیق با مدل های مبدل (**Transformers**) برای بازنمایی متن آشنا شویم.

Hamidreza Baradaran Kashani

نکته پایانی

- به بیان ساده، زبان یک مجموعه ثابت از قواعد است (البته با تعداد زیادی استثنا).
- هیچ چیزی در توکن بندی جهت مدلسازی و یادگیری وجود ندارد!
- بسیاری از استثناها در یک زبان وجود دارند که برای شناخت آنها قرار نیست از یادگیری ماشین استفاده شود.
- به بیان دیگر یادگیری ماشین برای کشف و شناسایی الگوها است. در حالیکه استثناهای یک زبان کاملا برخلاف الگوها (منظم) هستند.

- بنابراین برای توکن بندی نیازی به یادگیری ماشین نداریم بلکه تنها یک سری قواعد نیاز داریم که تمام استثناهای زبان را در نظر بگیرید.

Hamidreza Baradaran Kashani

ایست واژه ها

(Stopwords)

ایست واژه ها

- کلمات بسیار پرتکرار و رایج در یک زبان هستند و در اکثر متون ظاهر می شوند، مانند:
  - حروف تعریف، حروف اضافه و ربط، ضمایر، افعال کمکی و ....

- اطلاعات زیادی بخصوص از لحاظ معنایی ندارند.
  - مثال در زبان انگلیسی: **“the”, “and”, “or”, “is”, “it” , ...**
  - مثال در زبان فارسی: "و"، "یا"، "در"، "آن" و ....

- هیچ لیست کامل و دقیقی از ایست واژه ها در ابزارهای **NLP** وجود ندارد.
- در بسیاری از موارد حذف ایست واژه ها، باعث افزایش دقت تحلیل ها، تمرکز بیشتر بر روی کلمات مهمتر و کاهش حجم پردازش می شود.

There is a **book** on the **table**

Hamidreza Baradaran Kashani

ایست واژه ها

آیا همواره و در هر کاربردی در NLP باید ایست واژه ها را حذف کرد؟

- **قاعده کلی و دقیقی نداریم. این مساله بسیار وابسته به کاربرد و مساله مورد نظر است. در برخی موارد حذف آنها بسیار سودمند و در برخی موارد مضر است.**

- **مفید برای دسته بندی متن**
  - **با حذف ایست واژه ها، تمرکز بیشتر بر روی کلمات مهمتر و متمایزتر برای جداسازی دسته های مختلف متن**

- **مضر برای کاربردهایی مانند ترجمه ماشینی، پاسخ به سوالات و خلاصه سازی متن**
  - **در ترجمه ماشینی حتما این کلمات باید از زبان مبدا به مقصد ترجمه شوند.**
  - **در پاسخ به سوالات حذف آنها مثلا از سوال می تواند باعث تغییر معنا و مفهوم سوال شود.**
  - **در خلاصه سازی متن، باعث از بین رفتن ساختار گرامری و همین طور معنای متن خلاصه شده می شود.**

ایست واژه ها

❖مثال: با هدف کاربرد آنالیز احساسات

“The movie was not good at all”

❖ پس از حذف کلمات توقف:

“movie good”

❖ مشاهده می شود که احساس جمله پس از حذف ایست واژه ها بطور کامل تغییر کرده است.

❖ کتابخانه های مختلف شامل لیست ایست واژه ها:

❖ **NLTK, spaCy, Gensim, Scikit-Learn**

نرمالسازی کلمات و ریشه‌یابی

(Word Normalization and Stemming)

Hamidreza Baradaran Kashani

نرمالسازی

- نرمالسازی متن یعنی تبدیل آن به فرم استاندارد
- به عبارتی اگر یک کلمه چندین فرم مختلف دارد، با استفاده از نرمالسازی، تمام آنها به فرم یکسانی تبدیل می شوند.
  - مثلا labeled/lebelled یا extra-linguistic/extralinguistic/extra linguistic
- بنابراین نیاز به راهکارهایی برای هم ارز کردن دو کلمه داریم.

Hamidreza Baradaran Kashani

نرمالسازی

❖ **یک کاربرد مهم نرمالسازی: بازیابی اطلاعات**

- **می خواهیم عبارات جستجو (query terms) و متن نمایه شده (indexed text) فرم یکسانی داشته باشند، مثلا یکسان بودن U.S.A با USA**

- **مثلا عبارت زیر وارد شده است و آنچه که هدف جستجوی ما می باشد، بصورت زیر است:**
  - Enter: ***window*** Search: ***window***, ***windows***
  - Enter: ***Windows*** Search: ***Windows***

نرمالسازی

❖ **یک کاربرد مهم نرمالسازی: بازیابی اطلاعات (ادامه)**

- در بازیابی اطلاعات بسیار متداول است که تمام حروف بزرگ به حروف کوچک تبدیل شوند، چرا که بیشتر افراد با حروف کوچک جستجو را انجام می‌دهند.
- البته استثناهایی وجود دارند: استفاده از فرم حروف بزرگ در میانه جملات
  - ***General Motors***
  - ***Fed*** *vs.* ***fed***
  - ***SAIL*** *vs.* ***sail***
  - ***US*** *vs.* ***us***

❖ در کاربردهایی مثل تحلیل احساسات، استخراج اطلاعات و ترجمه ماشینی فرم حروف کوچک و حروف بزرگ اهمیت زیادی دارد.

Hamidreza Baradaran Kashani

ریشه یابی (Stemming)

- به طور کلی جستجو کردن در بخش‌هایی از کلمات (part of words) را در مبحثی با عنوان ریخت‌شناختی یا morphology بررسی می‌کنند.

- ریخت‌شناختی به مطالعه واژک یا morpheme می‌پردازد.
- واژک (morpheme)
  - کوچکترین واحد زبانی سازنده کلمه است و متشکل از دو نوع است:
    - Stems: جزء اصلی یک کلمه بوده که معنای اصلی کلمه را شامل می شود.
    - Affixes : اجزای دیگر کلمه که به stem متصل می‌شوند و اغلب کارکردهای گرامری دارند

Hamidreza Baradaran Kashani

ریشه یابی (Stemming)

- ریشه یابی به عبارت ساده به معنای زدن شاخ و برگ کلمات است.
- فرآیند Stemming یا ریشه‌یابی، به فرآیند استخراج Stem ها از کلمات با حذف یا جایگزینی Suffix ها اطلاق می‌شود.
- فرآیند Stemming وابسته به زبان می‌باشد و در زبان‌های مختلف روش‌های مختلفی برای آن وجود دارد.
  - e.g., ***automate(s), automatic, automation ---> automat***

الگوریتم Porter's Stemmer

❖ الگوریتم ریشه یاب **Porter** یکی از الگوریتم های ساده و درعین حال شناخته شده ریشه یابی در زبان انگلیسی است که از چند قاعده ساده متوالی برای دستیابی به **Stem** کلمات بهره میبرد.

Step 1a

```
sses → ss    caresses → caress
ies  → i     ponies   → poni
ss   → ss    caress   → caress
s    → ø     cats     → cat
```

Step 1b

```
(*v*)ing → ø  walking   → walk
               sing     → sing
(*v*)ed  → ø  plastered → plaster
…
```

Step 2 (for long stems)

```
ational→ ate relational→ relate
izer→ ize    digitizer → digitize
ator→ ate    operator  → operate
…
```

Step 3 (for longer stems)

```
al    → ø  revival    → reviv
able  → ø  adjustable → adjust
ate   → ø  activate   → activ
…
```

Hamidreza Baradaran Kashani

لم‌سازی (Lemmatization)

- لم‌سازی تبدیل فرم‌های مختلف یک کلمه به فرم پایه (**Base form or Headword**)
  - *am, are, is → be*
  - *car, cars, car's, cars' → car*

- *the boy's cars are different colors → the boy car be different color*

- در واقع در لم‌سازی **Headword** مربوط به هر کلمه از دیکشنری پیدا می‌شود.

Hamidreza Baradaran Kashani

تقطیع جملات

(Sentence Segmentation)

Hamidreza Baradaran Kashani

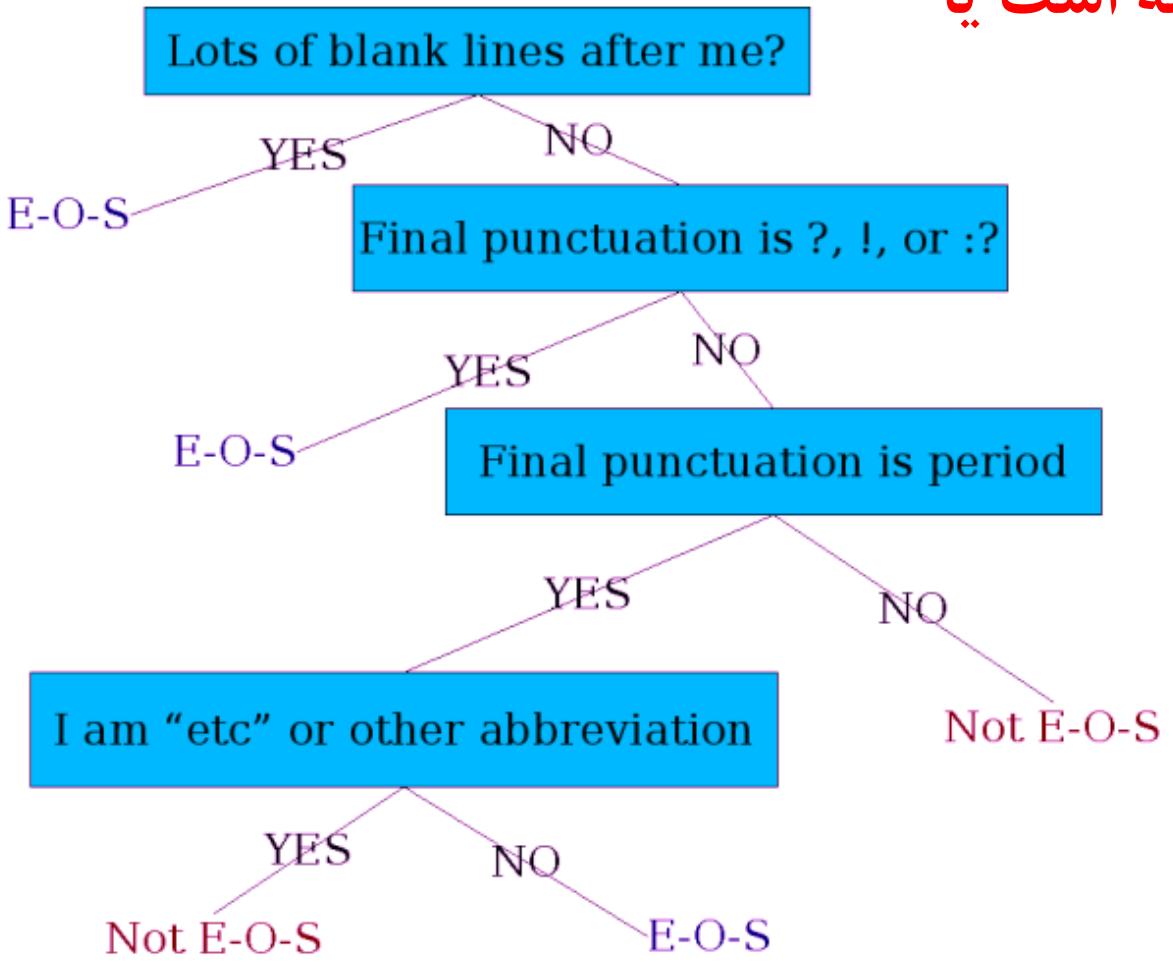
جداسازی جملات

- ؟ و ! علائم تقریبا غیر مبهم برای بیان انتهای جملات هستند.
- علامت نقطه (.) کاملا مبهم است، چرا که:
  - بسیاری از اختصارات با نقطه بیان می شوند مثلا Dr. یا U.S.A یا ...
  - اعدا اعشاری از نقطه استفاده می کنند: 0.2 یا 3.4 یا ...
- یک راه حل کلی: استفاده از طبقه‌بندی کننده (Classifier) با دو کلاس:
  - کلاس۱) انتهای جمله (EOS: End Of Sentence)
  - کلاس۲) نبودن انتهای جمله (Not EOS)
  - انواع طبقه‌بندی کننده ها: قواعد دستی (Hand-written Rules)، عبارات منظم (Regular Expressions) و یادگیری ماشین (Machine Learning)

Hamidreza Baradaran Kashani

درخت تصمیم: برای تعیین انتهای جمله

آیا یک کلمه در انتهای جمله است یا خیر؟



Hamidreza Baradaran Kashani

درخت تصمیم با ویژگی‌های پیچیده‌تر

❖ استفاده از ویژگی‌های زیر:

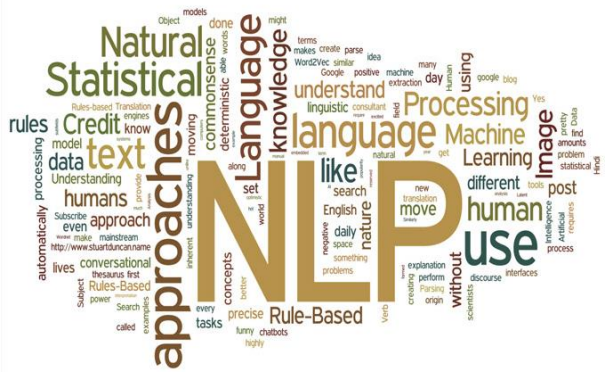
- شکل کلمه با نقطه و همچنین شکل کلمه پس از نقطه:
  - ابتدای آن با حرف بزرگ یا کوچک نوشته شده است: Uppercase یا lowercase
  - تمام حروف بزرگ هستند (Capital letters)
  - اعداد
- ویژگی های عددی مثل:
  - طول کلمه با نقطه (معمولا اختصارات طول کوتاهی دارند)
  - احتمال وقوع کلمه با نقطه در انتهای جمله
  - احتمال وقوع کلمه پس از نقطه در ابتدای جمله

Hamidreza Baradaran Kashani

انواع طبقه‌بندی کننده‌ها

- SVM (Support Vector Machines)
- Decision Trees
- Logistic Regressions
- Neural Nets
- etc.

Hamidreza Baradaran Kashani



با تشکر از توجه شما