Language Models for Understanding and Generation

Andrea Zugarini

https://sailab.diism.unisi.it

https://andreazugarini.github.io/

January 20, 2021

Universities of Florence and Siena, SAILab

Introduction

There have been tremendous progresses of Natural Language Processing (NLP) in the last decade.

Deep Learning brought dramatic improvements in almost any NLP task, ranging from understanding up to language generation.

But is Deep Learning the only reason behind such breakthroughs? **No**!

In this seminar, I will show how Language Modeling is crucial in the development of state-of-the-art models for NLP.

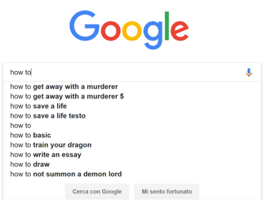
Language Modeling

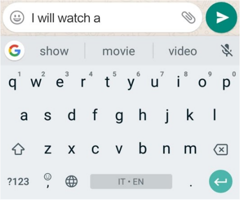
Language Modeling

Language Modelling is the problem of estimating the probability distribution of text.

Language Models are involved in many tasks and applications:

- Automatic Speech Recognition
- Spell Correction
- Word Suggestion





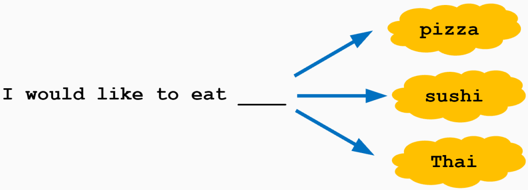
But, Language Modeling has become essential because it allows to **learn** powerful general purpose models for NLP.

Definition

Let be $\boldsymbol{w} := (w_1, \ldots, w_n)$ the words (or other tokens) of a text.

$$p(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} p(w_i | w_{i-1}, \ldots, w_1)$$

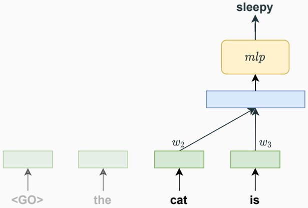
Language models estimate $p(w_i | w_{i-1}, \ldots, w_1)$ (or some approximations), i.e. they learn to predict which word comes next:



Neural Language Models

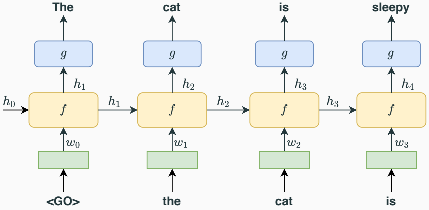
Neural Language Models estimate $p_\theta(w_i | w_{i-1}, \dots, w_1)$ with neural networks[1].

MLPs



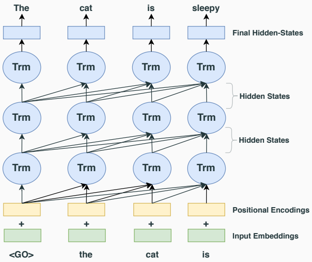
$$p_\theta(w_t | w_{t-1}, \dots, w_{t-k}), k = 2$$

Recurrent Neural Networks



$$p_\theta(w_t | h_t), h_t = f(w_{t-1}, h_{t-1})$$

Transformers



$$p_\theta(w_t | h_t), h_t = f(w_1, \dots, w_{t-1})$$

[1] Yoshua Bengio et al. "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155; Tomáš Mikolov et al. "Recurrent neural network based language model". In: *Eleventh annual conference of the international speech communication association*. 2010; Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

Evaluation

Language Models are usually evaluated in terms of perplexity.

Given a corpus $\mathcal{D} := (w_1, \ldots, w_N)$, $p_\theta(w_i | w_{i-1}, \ldots, w_1)$ the learnt distribution, the perplexity pp of p_θ in $\mathcal{D}$ is defined as:

$$pp(\mathcal{D}, p_\theta) := 2^{\frac{1}{N}\sum_{i=1}^{N} \log(p_\theta(w_i | w_{i-1}, \ldots, w_1))}.$$

The lower pp is, the better is the language model p_θ in $\mathcal{D}$.

Language Understanding

Language Representation

Representing Language is the first, essential step for any Language Understanding system.

Language is purely symbolic, whereas Machine Learning techniques are designed for sub-symbolic inputs.

To represent language in order to feed it into Machine Learning algorithms, we need to:

- Convert text into a sequence of symbols (**tokenization**);
- Assign a sub-symbolic representation to each symbol (**embedding**).

Tokenization

- Word-based: separates text into a sequence of words.
- Character-level: converts the string into a sequence of characters.
- Byte Pair Encoding (BPE)[2], WordPiece[3] and SentencePiece[4], Syllables tokenizers are trade-off strategies in between character-level and word-level splits.

`The cat sleeps`

Words: `[The, cat, sleeps]`

Characters: `[T, h, e, <s>, c, a, t, <s>, s, l, e, e, p, s, <s>]`

WordPiece: `[The, cat, sl, ##e, ##eps]`

---

[2] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909* (2015).

[3] Yonghui Wu et al. "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (2016).

[4] Taku Kudo and John Richardson. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing". In: *arXiv preprint arXiv:1808.06226* (2018).

Learning Language Representations

Embeddings are dense representations of the obtained tokens.

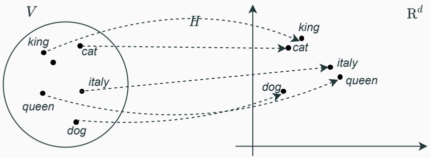
Given a set V of symbols, we define a function $H : V -> \mathrm{R}^d$ to assign a dense representation to each symbol.

- H is implemented as a matrix $E \in \mathrm{R}^{|V| \times d}$, known as Embedding matrix.
- $d << |V|$.
- If symbols are words: Word Embeddings (WEs).
- $|V|$ is large, between tens of thousands up to few millions.

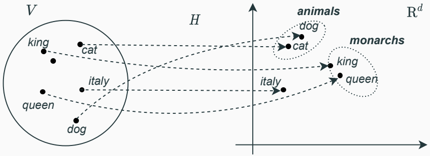
But how to map text into R^d? Random associations will perform poorly.

Language modeling allows to **learn** meaningful embeddings!

Random Embedding



Meaningful Embedding



Word Embeddings - CBOW & Skip-gram[5]

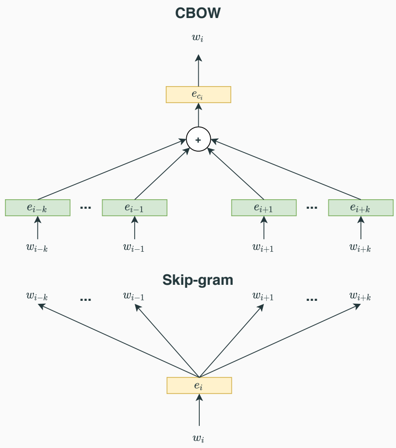
"You shall know a word by the company it keeps"

Estimate the probability of a word given its (left and right) context:
$p_\theta(w_i|c_i), c_i = (w_{i-k}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+k})$

CBOW:

- $e_i = \sum_{j=i-k, j\neq i}^{i+k} e_j$
- $p_\theta(w_i|c_i) = \texttt{softmax}(e_i)$

Skip-gram is the dual version of CBOW.



[5] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

Word Embeddings - Limitations

WEs lack of morphological information about text:

- Crucial in specific use-cases.
- Overcomes the problem of unknown and rare tokens.
- Can reduce dramatically the vocabulary size.

Multi-sense words have a unique representation:

- Actual meaning of a word highly depends on the context in which it is placed.
- Having contextual representations of text is essential and improves performances in any language understanding problem.

Character-aware Word and Context Representations[7]

PROPOSAL:

We present a character-aware neural language model that overcomes limitations of word-based embeddings.

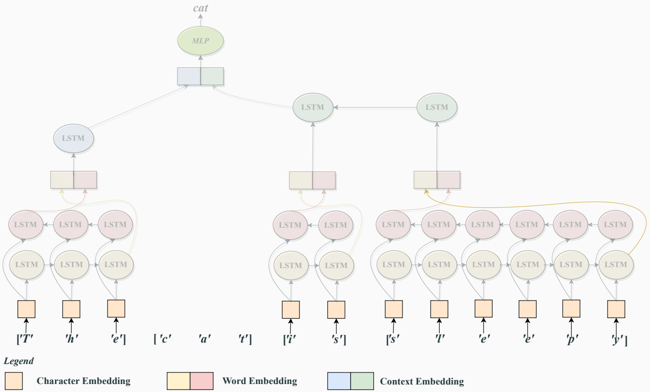
The model effectively learns representations of words and contexts, with an unsupervised learning mechanism that follows the same principle of CBOW and *context2vec*[6].

[6] Oren Melamud, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 2016, pp. 51–61.

[7] Giuseppe Marra et al. "An unsupervised character-aware neural approach to word and context representation learning". In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 126–136.

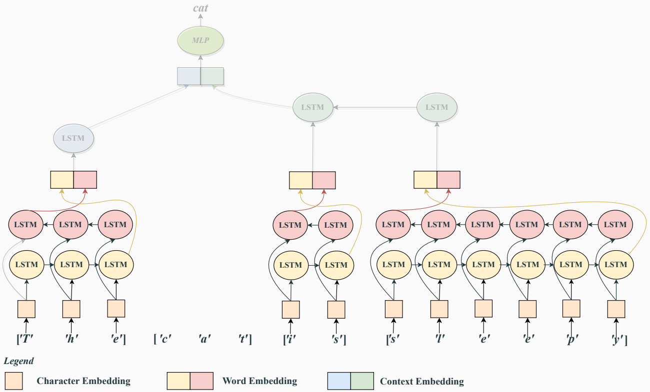
Encoding Characters

Text is tokenized in sequences of words. Each word is further split in a sequence of characters. Each character is associated to its embedding:



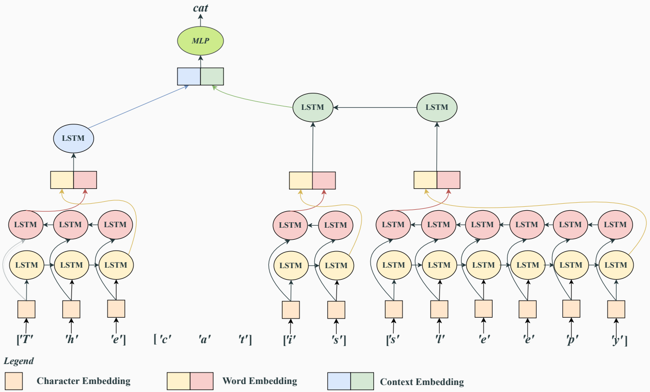
Encoding Words

A bi-directional LSTM encodes each word by processing its sequence of characters *forward* and *backward*:



Encoding Contexts

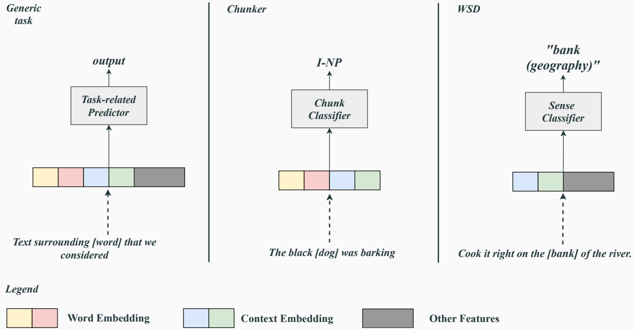
On top of the embedded words, another bi-LSTM encodes contexts:



The model allows to construct also *contextual* representations of a word, opportunely selecting the bi-LSTM states that include the current word itself.

Learned Representations - Usage

The encoder can be used as a features extractor:



Trained on ukWaC[8] (2 billion words).

[8] http://wacky.sslmit.unibo.it/doku.php?id=corpora

Experiments - Chunking

Dataset: **CoNLL 2000**, a standard benchmark, containing 23 classes.

Classifier: Bi-LSTM fed with both Word and Context embeddings.

Input Features	F1 %
Our WE only	89.68
Our CE only	89.59
Our WE + Our CE	**93.30**
WE + CE Trained on Task	89.83

Model	F1 %
Collobert et al.	94.32
Huang et al.	**94.46**
Huang et al. – POS	93.94
Our model	93.30
Our model + POS	93.94

Note: Both competitors[9] use CRFs and POS features.

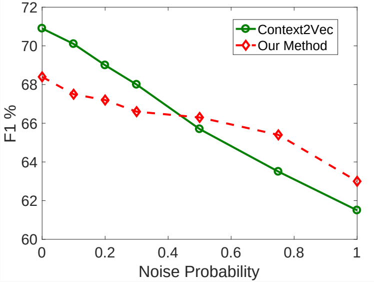
[9] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537; Zhiheng Huang, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging". In: *arXiv preprint arXiv:1508.01991* (2015).

Experiments - Word Sense Disambiguation

Classification with the traditional IMS approach[10] based on a SVM classifier on multiple benchmarks[11].

Model	SE2	SE3	SE2007	SE2013	SE2015	ALL
IMS	70.2	68.8	62.2	65.3	69.3	68.1
IMS+word2vec	72.2	69.9	62.9	66.2	71.9	69.6
IMS+context2vec	**73.8**	**71.9**	**63.3**	**68.1**	**72.7**	**71.1**
IMS+Our CE	72.8	70.5	62.0	66.2	71.9	69.9

Our encoder has about **16 times less** trainable parameters than *context2vec*.



[10] Zhi Zhong and Hwee Tou Ng. "It makes sense: A wide-coverage word sense disambiguation system for free text". In: *ACL*. 2010, pp. 78–83.

[11] Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. "Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison". In: *Proc. of EACL*. 2017, pp. 99–110.

Natural Language Generation

Text Generation is Language Modeling

Natural Language Generation (NLG) problems can be formulated as special instances of Language Modeling.

Let us divide $\boldsymbol{w}$ in two disjoint sequences $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_m)$, where $\boldsymbol{x}$ are given and $\boldsymbol{y}$ has to be generated:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{m} p(y_i|y_{<i}, \boldsymbol{x}),$$

Task	$\boldsymbol{x}$	$\boldsymbol{y}$
Machine Translation	Source Language	Translated Text
Paraphrasing	Text	Paraphrase
Text Summarization	Article/Paragraph	Summary
Language Modeling	$\emptyset$	Any Text

Open vs non-open ended text generation[12]

We distinguish among two kinds of text generation:

Open-ended

- Story Generation
- Text Continuation
- Poem Generation
- Lyrics Generation
- ...

Non open-ended

- Machine Translation
- Text Summarization
- Text Paraphrasing
- Data-to-text generation
- ...

Open and non-open ended models are trained in the same way.

At inference time however, different decoding strategies are required, depending on the type of generation problem.

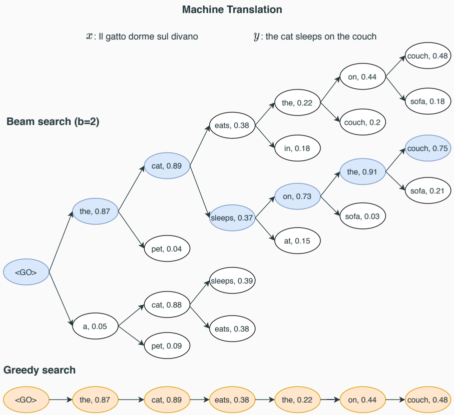
[12] Ari Holtzman et al. "The curious case of neural text degeneration". In: *arXiv preprint arXiv:1904.09751* (2019).

Decoding Strategies - Likelihood Maximization

Goal: Find the most probable sequence $\boldsymbol{y}$ given $\boldsymbol{x}$ from p_θ.

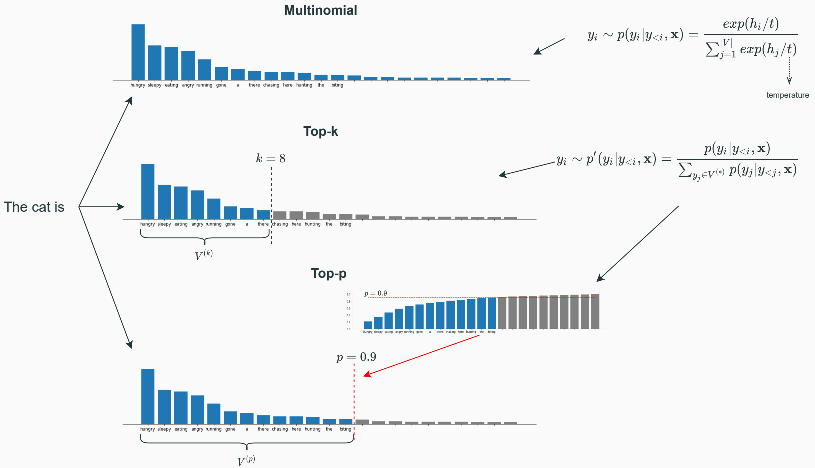
$$\boldsymbol{y} = (y_1, \ldots, y_m) = \arg\max_{\boldsymbol{y}} \prod_{i=1}^{m} p_\theta(y_i | y_{<i}, \boldsymbol{x})$$

- Unfortunately, finding the optimal $\boldsymbol{y}$ is intractable.
- Therefore, Search methods that explore only a small subset of sequences have been devised.
- *Beam* and *greedy search* are the most popular ones.
- Particularly effective for non-open ended tasks.



Likelihood maximization leads to poor, repetitive results in open ended problems.

Sampling produce diverse results.



Neural Poetry[13]

Poem Generation is a challenging problem, since:

- Poetry has unique features: structure, rhymes, meters and each author has their own style.
- The resources available are much poorer than other NLG problem, especially for ancient poetry.

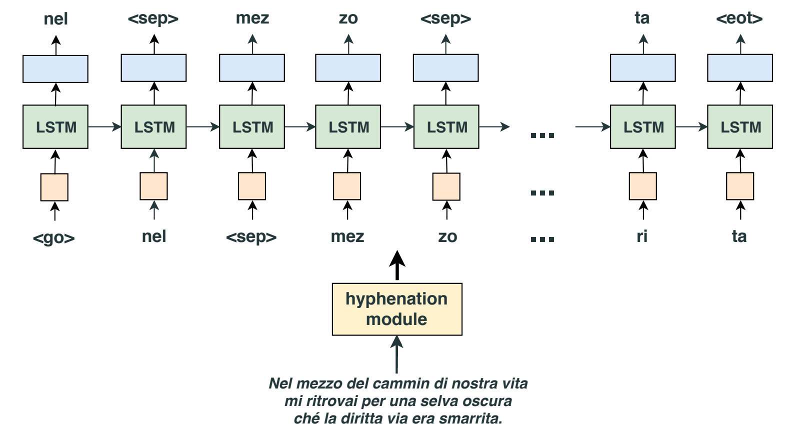
Proposal:

- Syllable-based LM allowing strong transfer learning from modern texts that is trained in a multi-stage fashion.
- A poem selection mechanism that is based on poem and author characteristics.

[13] Andrea Zugarini, Stefano Melacci, and Marco Maggini. "Neural Poetry: Learning to Generate Poems Using Syllables". In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 313–325.

Syllable Language Model (sy-LM)

- Hyphenation module: tokenizes input and output text into a sequence of syllables.
- Language model: At each time step t outputs $p_\theta(x_t|x_{<t})$.



Multi-stage Transfer Learning

Goal: Alleviate the problem of lack of available resources.

Idea: progressively grasp knowledge, from generic syntactical and grammatical information about the language itself, up to the author's style.

Role of syllables:

- At **syllable** level there are not many differences between poetic and **non-poetic** languages.
- Syllables have changed little in modern languages.

Generation Procedure

Once trained, **sy-LM** is exploited to generate new poems, with the following approach:

1. Generate N samples with Multinomial sampling from $p(x_t|x_1, \dots, x_{t-1})$.
2. Assign a score $R(x)$ to each generated sequence x.
3. Select the K sequences with highest score.

$R(x)$ is an average of four different functions aimed at scoring structure, meter, rhyme, lexicon of the tercet.

Experiments - Problem Setting

We focus on Dante Alighieri, the most important Italian Poet.

Data

- **DC**: Divine Comedy, 4811 tercets divided in train set (80%), validation set (10%) and test set (10%).
- **DP**: Other Dante's compositions, some of them are in prose.
- **PAISA'**: a large corpus of contemporary Italian texts.

Evaluation

- Performances using different training data sources.
- Human assessment of generated tercets from expert and non-expert judges.

Experiments - Multi-stage Transfer Learning

Perplexity on validation and test set, pre-training the model using multiple datasets.

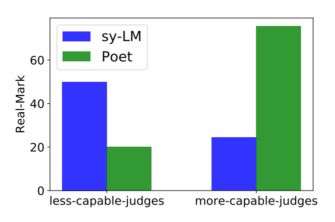
Datasets	Val PPL	Test PPL
DC	12.45	12.39
PAISA' $\rightarrow$ DC	10.83	10.82
DP $\rightarrow$ DC	11.95	11.74
PAISA' $\rightarrow$ DP $\rightarrow$ DC	**10.63**	**10.55**

A $\rightarrow$ B means that we train on A first, and then we train on B.

Experiments - Human Evaluation

Non-expert Judges

Generator	Real-Mark
sy-LM	28%
Poet	64%



Annotators' task: decide whether a tercet was real or not.

Expert Judges

	Readability	Emotion	Meter	Rhyme	Style
Judge 1	1.57	1.21	1.57	3.36	2.29
Judge 2	1.64	1.45	1.73	3.00	2.27
Judge 3	2.83	2.33	2.00	4.17	2.92
Judge 4	2.17	2.00	2.33	2.92	2.50
Average	2.04	1.73	1.90	3.37	2.49
Poet (Average)	4.34	3.87	4.45	4.50	4.34

Each expert evaluated 20 tercets, 10 generated and 10 real.

Experiments - Some Tercets

e tenendo con li occhi e nel mondo
che sotto regal facevan mi novo
che ’l s’apparve un dell’altro fondo

in questo imaginar lo ’ntelletto
vive sotto ’l mondo che sia fatto moto
e per accorger palude è dritto stretto

per lo mondo che se ben mi trovi
con mia vista con acute parole
e s’altri dicer fori come novi

non pur rimosso pome dal sospetto
che ’l litigamento mia come si lece
che per ammirazion di dio subietto

Analysis of Language Varieties

Language Models for Language Varieties

A language variety is a subcategory of a language: *dialects*, *idiolects*, *diachronic* languages.

Language models and perplexity can be used to provide a measure of similarity between language corpora[14].

Perplexity $pp(\mathcal{D}, p)$ is a function of the probability distribution p and the reference corpus $\mathcal{D}$.

Usually $\mathcal{D}$ is fixed, so that different language models are compared. Analogously, we can fix p and change the evaluation corpus.

[14] José Ramom Pichel Campos, Pablo Gamallo, and Iñaki Alegria. "Measuring language distance among historical varieties using perplexity. Application to European Portuguese.". In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018, pp. 145–155; José Ramom Pichel Campos, Pablo Gamallo Otero, and Iñaki Alegria Loinaz. "Measuring diachronic language distance using perplexity: Application to English, Portuguese, and Spanish". In: *Natural Language Engineering* 26.4 (2020), pp. 433–454.

Perplexity-based Language Measures

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two corpora and $\texttt{LM}_{\mathcal{L}_1}, \texttt{LM}_{\mathcal{L}_2}$ two language models trained on $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

Symmetric Perplexity-based Language Distance (**PLD**)[15]:

$$PLD(\mathcal{L}_1, \mathcal{L}_2) := \frac{pp_{\mathcal{L}_1 \to \mathcal{L}_2}(\mathcal{L}_2, \texttt{LM}_{\mathcal{L}_1}) + pp_{\mathcal{L}_2 \to \mathcal{L}_1}(\mathcal{L}_1, \texttt{LM}_{\mathcal{L}_2})}{2}.$$

Asymmetric indicator, Perplexity-based Language Ratio (**PLR**)[16]:

$$PLR(\mathcal{L}_1, \mathcal{L}_2) := \frac{pp_{\mathcal{L}_1 \to \mathcal{L}_2}(\mathcal{L}_2, \texttt{LM}_{\mathcal{L}_1})}{pp_{\mathcal{L}_2 \to \mathcal{L}_1}(\mathcal{L}_1, \texttt{LM}_{\mathcal{L}_2})}.$$

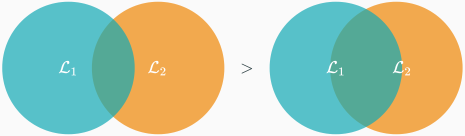
[15] Pablo Gamallo, José Ramom Pichel Campos, and Inaki Alegria. “A perplexity-based method for similar languages discrimination”. In: *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects (VarDial)*. 2017, pp. 109–114.

[16] Andrea Zugarini, Matteo Tiezzi, and Marco Maggini. “Vulgaris: Analysis of a Corpus for Middle-Age Varieties of Italian Language”. In: *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*. 2020, pp. 150–159.

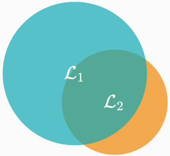
Perplexity-based Language Measures

Intuitively:

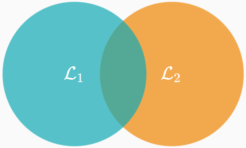
$PLD(\mathcal{L}_1, \mathcal{L}_2)$



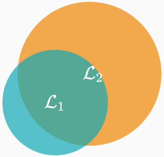
$PLR(\mathcal{L}_1, \mathcal{L}_2) < 1$



$PLR(\mathcal{L}_1, \mathcal{L}_2) = 1$



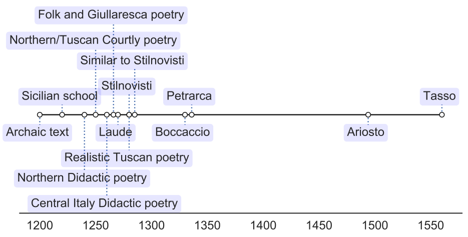
$PLR(\mathcal{L}_1, \mathcal{L}_2) > 1$



Vulgaris[18]

Collection of an heterogeneous literary text corpus[17] of Middle-Age Italian language:

- Time period of about four centuries.
- Text enriched with metadata such as style, properties, verse and stanza separators.
- Compositions are grouped into 14 families accordingly to stylistic and spatio-temporal features (see below):



[17] from `http://www.bibliotecaitaliana.it/`

[18] Zugarini, Tiezzi, and Maggini, "Vulgaris: Analysis of a Corpus for Middle-Age Varieties of Italian Language".

Analysis

Families grouped chronologically in four diachronic varieties: **XIII**, **XIV**, **XV-XVI-1**, **XV-XVI-2**.

	XIII	XIV	XV-XVI-1	XV-XVI-2
# words	455583	1480379	484276	1669928
dataset proportion (%)	11.14	36.19	11.84	40.83
# unique words	57343	73530	42594	72369
Avg occurrences per word	7.94	20.13	11.37	23.08

A character LM $p_\theta(x_i|x_{i-1}, \ldots, x_1, a, f, k)$ conditioned with external meta information - author, family and kind of composition (prose|poetry) - is trained on each variety.

Analysis - Perplexity-based Indicators

PLD is lower in diachronic varieties closer in time:

	XIII	XIV	XV-XVI-1	XV-XVI-2
XIII	3.90	5.38	5.99	6.08
XIV	5.38	3.52	4.76	4.65
XV-XVI-1	5.99	4.76	3.30	4.47
XV-XVI-2	6.08	4.65	4.47	3.28

PLR highlights a strong asymmetric behaviour on perplexity pairs involving the set XIII, due to the heterogeneity of the group:

	XIII	XIV	XV-XVI-1	XV-XVI-2
XIII	1.00	0.81	0.65	0.72
XIV	**1.23**	1.00	0.86	0.95
XV-XVI-1	**1.53**	1.16	1.00	1.14
XV-XVI-2	**1.39**	1.05	0.88	1.00

Thank you for listening!

Char-aware LM: `github.com/sailab-code/char-word-embeddings`
Neural Poetry: `gitlab.com/zugo91/nlgpoetry`
Language Varieties: `github.com/sailab-code/vulgaris`

Andrea Zugarini
https://andreazugarini.github.io/

@AZugarini

References

Bengio, Yoshua et al. “A neural probabilistic language model”. In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.

Campos, José Ramom Pichel, Pablo Gamallo, and Iñaki Alegria. “Measuring language distance among historical varieties using perplexity. Application to European Portuguese.”. In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018, pp. 145–155.

Campos, José Ramom Pichel, Pablo Gamallo Otero, and Iñaki Alegria Loinaz. “Measuring diachronic language distance using perplexity: Application to English, Portuguese, and Spanish”. In: *Natural Language Engineering* 26.4 (2020), pp. 433–454.

Collobert, Ronan et al. “Natural language processing (almost) from scratch”. In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537.

Gamallo, Pablo, José Ramom Pichel Campos, and Inaki Alegria. “A perplexity-based method for similar languages discrimination”. In: *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects (VarDial)*. 2017, pp. 109–114.

Holtzman, Ari et al. “The curious case of neural text degeneration”. In: *arXiv preprint arXiv:1904.09751* (2019).

Huang, Zhiheng, Wei Xu, and Kai Yu. “Bidirectional LSTM-CRF models for sequence tagging”. In: *arXiv preprint arXiv:1508.01991* (2015).

Kudo, Taku and John Richardson. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing". In: *arXiv preprint arXiv:1808.06226* (2018).

Marra, Giuseppe et al. "An unsupervised character-aware neural approach to word and context representation learning". In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 126–136.

Melamud, Oren, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 2016, pp. 51–61.

Mikolov, Tomas et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

Mikolov, Tomáš et al. "Recurrent neural network based language model". In: *Eleventh annual conference of the international speech communication association*. 2010.

Radford, Alec et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

Raganato, Alessandro, Jose Camacho-Collados, and Roberto Navigli. "Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison". In: *Proc. of EACL*. 2017, pp. 99–110.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909* (2015).

Wu, Yonghui et al. "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (2016).

Zhong, Zhi and Hwee Tou Ng. "It makes sense: A wide-coverage word sense disambiguation system for free text". In: *ACL*. 2010, pp. 78–83.

Zugarini, Andrea, Stefano Melacci, and Marco Maggini. “Neural Poetry: Learning to Generate Poems Using Syllables”. In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 313–325.

Zugarini, Andrea, Matteo Tiezzi, and Marco Maggini. “Vulgaris: Analysis of a Corpus for Middle-Age Varieties of Italian Language”. In: *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*. 2020, pp. 150–159.

Generation Procedure - Scoring Criteria

- STRUCTURE

$$R_1(x) = 1 - \mathtt{abs}(|x| - 3)$$

- METER

$$R_2(x) = \sum_{v \in x} 1 - (\mathtt{abs}(|v| - 11))$$

- RHYME

$$R_3(x) = \begin{cases} 1, & \text{if } (v_1, v_3),\ v_1, v_3 \in x \text{ are in rhyme} \\ -1, & \text{otherwise} \end{cases}$$

- VOCABULARY

$$R_4(x) = \sum_{w \in x} f_w(x), \qquad f_w(x_i) = \begin{cases} a, & \text{if } w \in V \\ -b, & \text{otherwise} \end{cases}$$