IACS Capstone Project Course

DataShack 2018

CROSS LINGUAL TEXT CLASSIFIERS

Srivatsan Srinivasan, Andrea Porelli, Alessandro Bianchi, Ginevra Terenghi

TRIBE DYNAMICS

POLITECNICO
MILANO 1863

IACS
INSTITUTE FOR APPLIED
COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

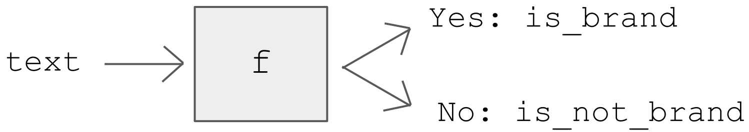
Who: Tribe Dynamics

Central Problem

What: How much people talk about a particular brand on social media

Why: So companies can asses popularity of their brand and products

How: Model



So what? They already have a model, how can we help to improve it?

Tribe Dynamics

Model

Supervised model used to **classify posts** (output probability of each being about a brand or not)

1) find the **function** that defines the boundary in input space dividing `is_brand` from `is_not_brand`

2) learn the **parameters** of the boundary

3) **maximize the correctness** of each post being about a brand or not (i.e. fitness)

So what? It is a lot of work! Plus, how should we convert text to a numerical format?

Convert Words to Vectors: Bag of Words...

Text

|

|--- Unique Words

|

|--- Count Occurrences

|

Vector

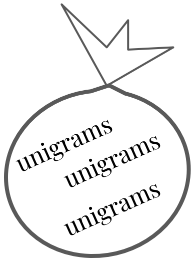
1 | "I love love this shampoo and I hate hate this soap!"

2 | "This shampoo is the best, it is super ."

Vocabulary

love, shampoo, hate, soap, is, best, super

1	2	1	2	1	0	0	0
2	0	1	0	0	2	1	0



Bag of Words

Each post is represented by one n-dimensional vector where n is the number of unique words in the entire dataset. What happens if we add a new word? a) longer vector, b) train the model again

... and N_grams

1 | "I love love this shampoo and I hate hate this soap!"

2 | "This shampoo is the best, it is super ."

Text

|

|---   Unique N_grams

|

|---   Count Occurrences

|

Vector

Bi-grams Vocabulary

love love, **love shampoo**,
shampoo hate, hate hate,
, shampoo is, is best, best is,
is super

[1, 1, 1, 1, 1, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1, 1, 1, 1]

Dim. bi-grams = $\binom{V}{2}$; Vocabulary = $O(V^n)$

Observation: a) vector size increases exponentially with n and polynomially with V (O(V^n)). and b) increasing zeros in the vector (i.e. sparse)

Tribe Dynamics

Pitfalls

So what? A supervised model that classifies brands presence from posts is hard because:

a) To learn a good classifier function:

   i) is hard to compute parameters of sparse input vectors

   ii) is computationally intensive to use iterative maximization algorithm (differentiation or matrix inversion in closed form).

b) To maintain both the model and the training set:

   i) must fit a new model to add new words

   ii) is expensive to label posts

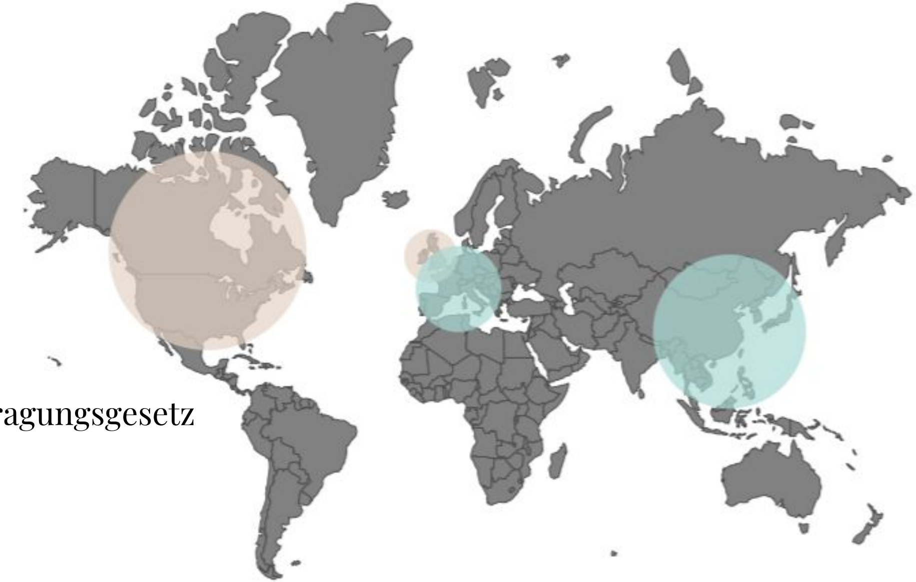
Even harder: monolingual to multilingual models

Not all languages can be reduced to unigrams so easily. Some have words that include multiple concepts.

Ex.

Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz

"the law for the delegation of monitoring beef labeling"



How can we build a model that performs posts' classification and is correct enough across different languages?

Classification Task Performance

Tribe current performances on **Non-English** languages, when classified using an n-grams logistic regression model on bag of multilingual vocabulary.

Our team **replicated Tribe's model** and we found similar results.

Imbalance : English has more data and datasets are significantly imbalanced across languages (measure AUC/PR/F1, not accuracy)

N-grams Logistic Regression Model

[AUC]

	Tribe	**Our Team**
English :	**0.95**	**0.96**
Non English:	**0.76**	**0.82**

Observation : Ground Truth errors (approximately 10-15% in non-English) infuses less confidence in models.

First Model Proposal

Model	Pros	Cons
N_grams: *"a troubled love story"*	interpretable	expansion of vocabulary
Word Embeddings + Alignment: *"one model forever"*	compact representation	difficult to learn params

↓

sparse input vector	**dense input vector**
[1, 1, 1, 0, 1, ... , 0, 1 0, 0]	[0.2, 0.1, 0.5, ... , 0.1, 0.3]
dimension is not fixed	dimension is fixed
change with more words	do not change with more words

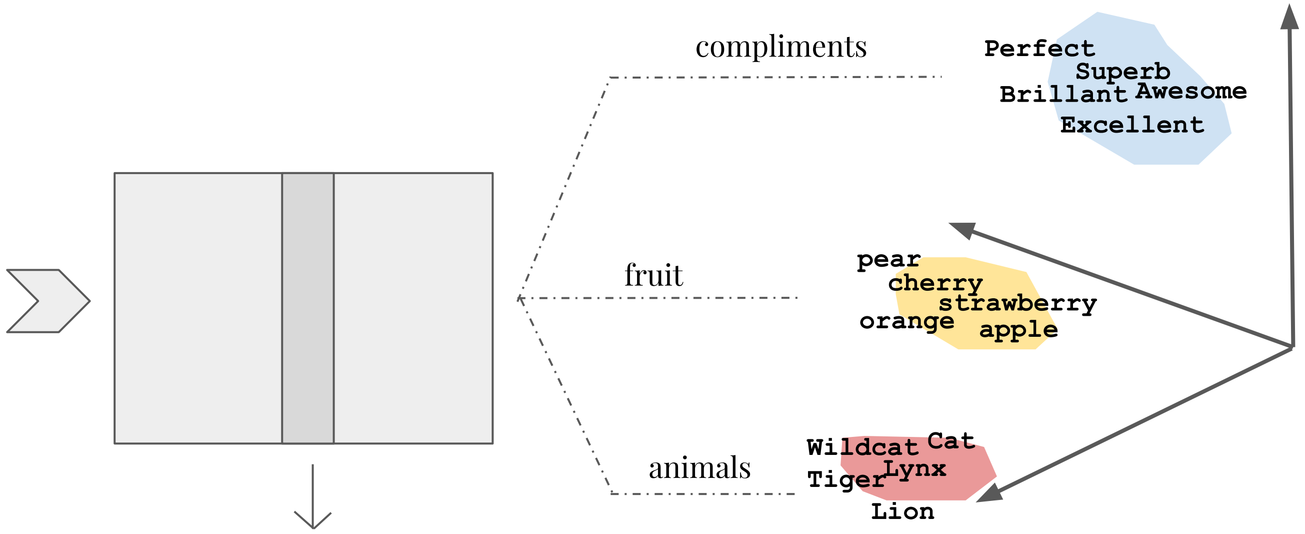
Mono-Lingual Word Embeddings: Intuition

INPUT:
[Monolingual Vocabulary*]

MODEL:
[Neural Network]

OUTPUT:
[Words close by meaning]

Word	Vector
Perfect	[1, 0, ...]
Pear	[0, 1, ...]
Cat	[0, 0, ...]
Superb	[...]
Cherry	[...]
...	[...]



* sample total population

hidden layer: embedding

Word Embeddings Training : Language Model

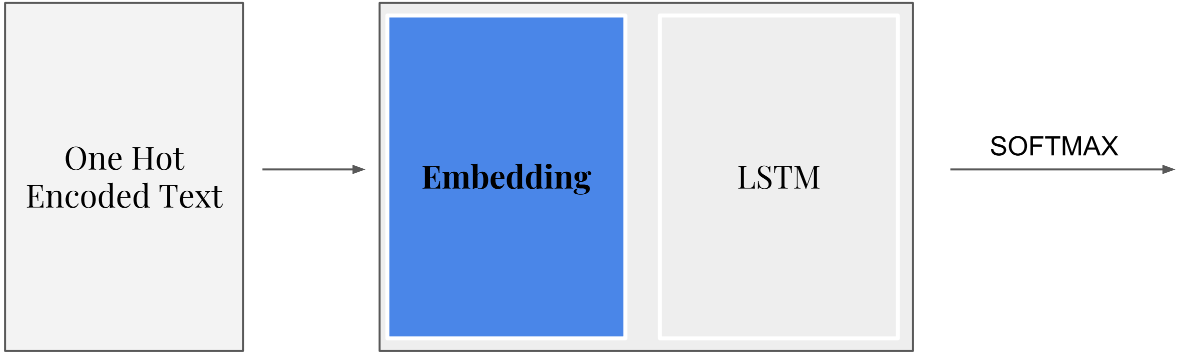
INPUT:

[Dogs, are]   [Cats, are]

MODEL:

[Neural Network]

SAMPLE OUTPUTS:



cute - 0.4

scary - 0.2

annoying - 0.1

lovable - 0.3

- Unsupervised
- Context encoded in sentence
- Similar words » Similar successors

Tribe Dynamics

Proposal I : Static and Dynamic Channel Embeddings

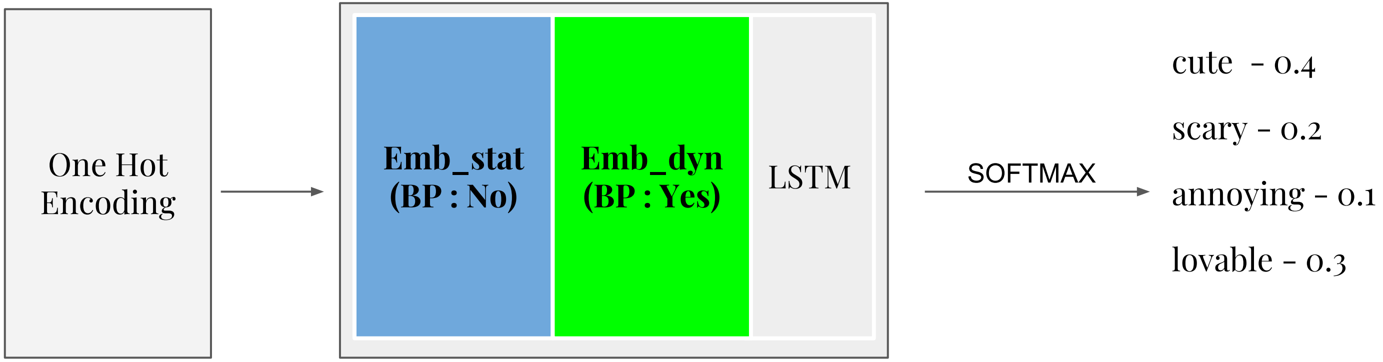
INPUT:

[Dogs, are] [Cats, are]

MODEL:

[Neural Network]

SAMPLE OUTPUTS:



Motivation: Standard knowledge (Wiki) - Static, Fashion Knowledge - Dynamic

Mono-Lingual Word Embeddings: evaluation process and results

Baseline: Bag of Words + Logistic Regression

|--- uni and bigrams (N dimensional)

|--- classifier: is_brand or is_not_brand

|--- estimate ROC, AP, F1

Model 1: Word Embeddings + Logistic Regression

|--- embedding output (300 dimension)

|--- input: sum of word vectors in the post

|--- classifier: is_brand or is_not_brand

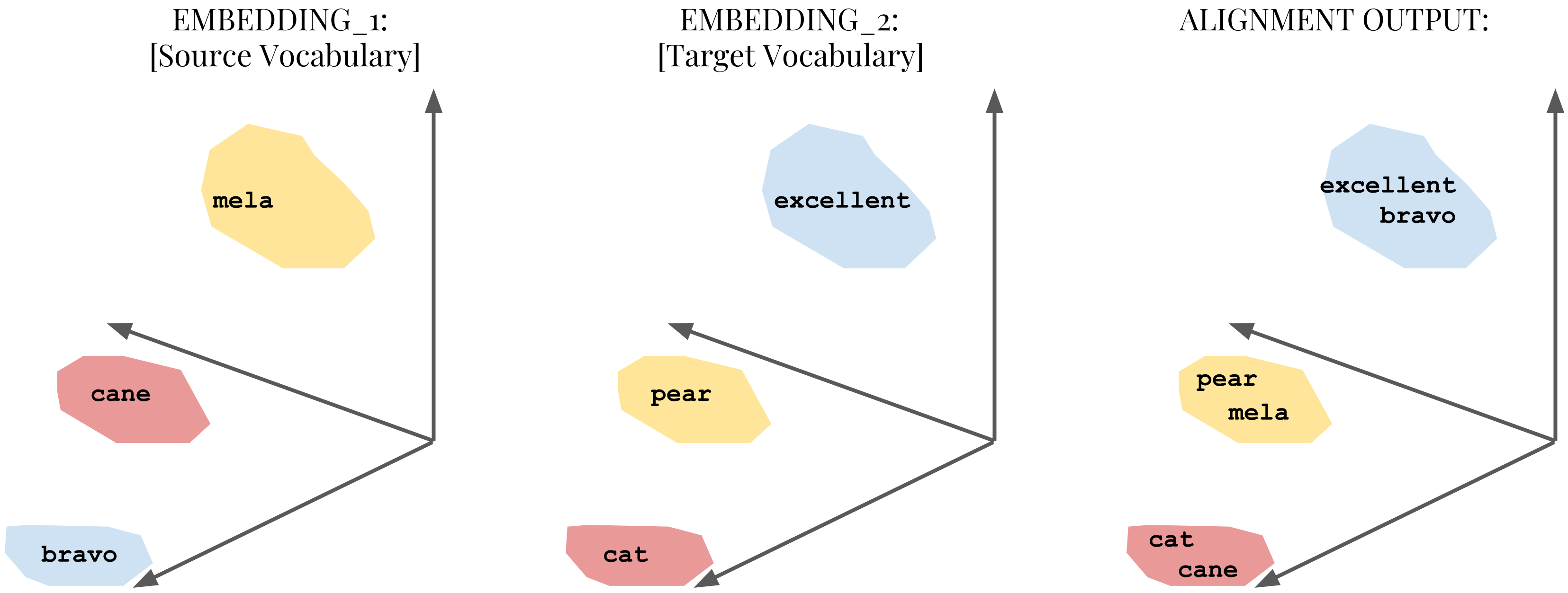
|--- estimate ROC, AP, F1

	[ROC]		[AP]		[F1]	
	BOW	**Emb**	**BOW**	**Emb**	**BOW**	**Emb**
English :	0.88	0.86	0.98	0.97	0.96	0.96
Italian:	0.84	0.85	0.90	0.90	0.86	0.86

BOW: Bag-of-Words (n-grams = (1,2)) + Logistic Regression
Emb: Word Embeddings + Logistic Regression

Tribe Dynamics

Multi-Lingual Aligned Word Embeddings



Alignment

process

|---   bilingual vocabulary (anchor words)

|---   input: source and target embeddings

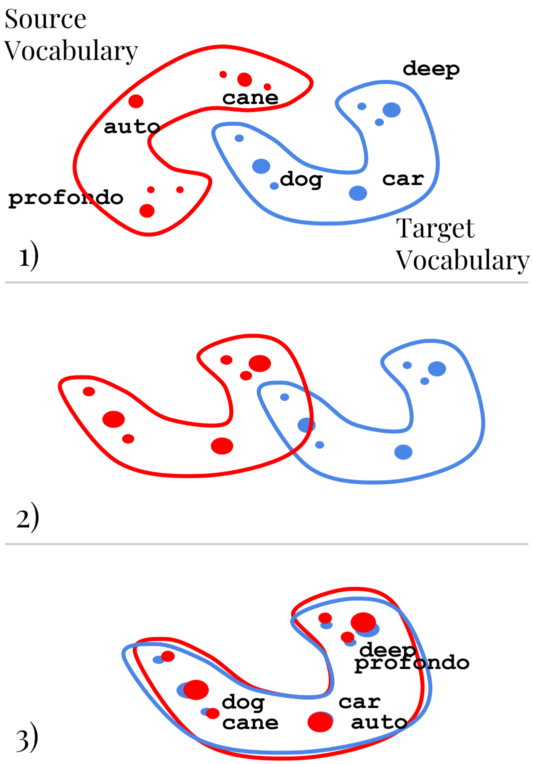
|---   learn transformation

|---   fitness: minimize euclidean distance of anchor words

|---   apply transformation

|---   estimate: euclidean distance, cosine similarity

* transformations W* aligns source and target vocabularies using rotation (1-2) and translation (2-3)



Advantage: only one model must embedded in vector space (knowledge transfer)

Tribe Dynamics

Alignment Definition

|--- is a linear distance-preserving map (isometry):

| f : X -> Y

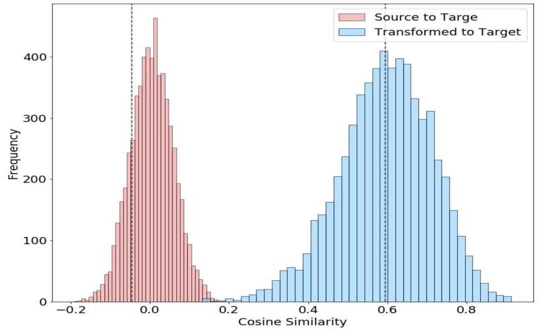
| $||f(x)|| = ||x||$; for all x in X

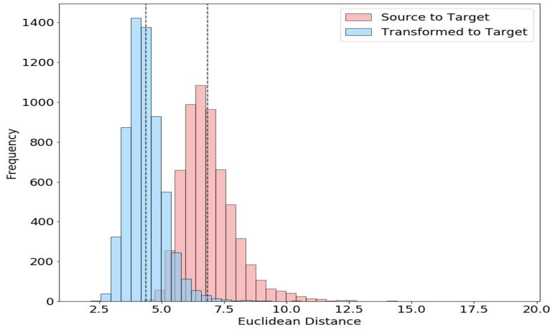
|

|--- objective function minimizing the euclidean distance:

$$W^* = \underset{W \in O_d(\mathbb{R})}{\arg\min} ||WX - Y||_F = UV^T, \quad \text{with} \quad U\Sigma V^T = \text{SVD}(YX^T)$$

orthogonal Procrustes problem





Multilingual Word Embeddings Aligned: evaluation process and results

Bag of Words + **Logistic Regression**

|--- same procedure as before

Aligned Word Embeddings + **Logistic Regression**

|--- embedding output

|--- align source embedding to target

|--- input: sum of word vectors in the post (source transformed and target vocabularies*)

|--- classifier on both languages: is_brand or is_not_brand

|--- estimate ROC, AP, F1

	[ROC]		[AP]		[F1]	
	BOW	Emb	BOW	Emb	BOW	Emb
English+Italian*:	0.89	0.85	0.99	0.97	0.96	0.95
Italian+English*:	0.89	0.86	0.99	0.97	0.96	0.94

BOW: Bag-of-Words (n-grams = (1,2)) + Logistic Regression
Emb: Aligned Word Embeddings + Logistic Regression

* use target embedding vector if two words are the same

Practical Discussion on Aligned Embeddings

1. **One-time cost** - New language train and align embeddings once. Retrain infrequent.
2. **Knowledge Transfer -** Happens at the embedding level.
3. **Singular Model -** One single model across languages.
4. **Generalizable -** Works with any classification model on embedded space.

   Eg: Logistic Regression, Feedforward nets, ConvNets etc.
5. **Data Boost -** Number of effective data points for model training increases.

Proposal 2 : Mixture Model

Why do we need two models?

Deep Learning Training is hard. Simpler model, alternative philosophy -

What if?

One model in English. Translate all other languages to English.

Advantages: Maintenance, Core Strengths, Lesser n-grams, Lesser data costs in non-English

Issues

- Sentence Level: Inaccurate, Costly
- Word Level: Word Sense Disambiguation

Word Sense Disambiguation

Key Idea: Context drives translation of words across languages.

EXAMPLE

Dove sei? (Not brand) - Where are you?

Io amo Dove (brand) - I love Dove.

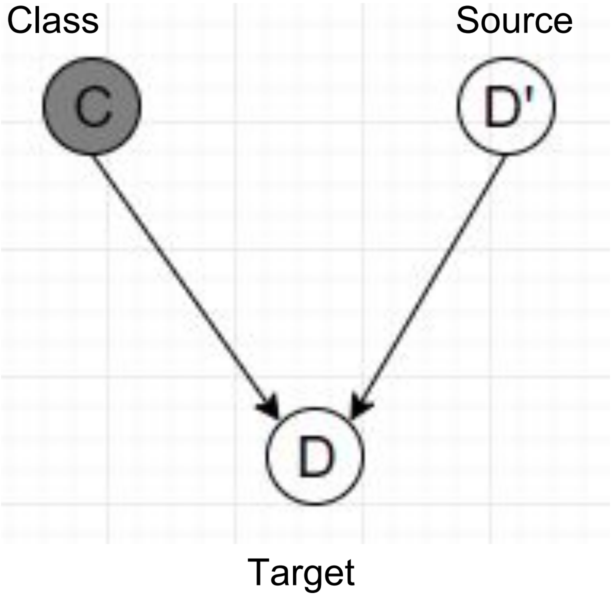
MAIN IDEA

- Each word has a mixture of translations.

- Infer the correct translation through context during training.

Conditional Translations - PGM

- Generative model
- Document - Bag of words
- P(D|D’): “**amo Dove”** - “love where”, “love Dove” (50%)
- P(D|D’, C=brand): “**amo Dove”** - “love Dove” (>>50%)
- Needs bilingual lexicon and EM



Source Classification and Class Inference

$$\widehat{c} = argmax_{c \in C} \prod_{i=1}^{v} \sum_{j=1}^{n_i} P(w_t^{ij} | w_s^i, c) e^{\lambda_{w_s^i} f(w_t^{ij}, c)}$$

Conditional Translation

Classifier in source (Softmax)

IDEA

- Target class maximizes likelihood of generation
- Could be viewed as weighted ensemble prediction by different translations to source language

RESULTS (ITALIAN)

Metric	B1	B2	B3	MM	MM + Sup*
Accuracy	0.81	0.87	0.95	0.91	0.92
AUC	0.58	0.66	0.83	0.74	0.74
PR	0.08	0.34	0.62	0.55	0.56
F1	0.17	0.52	0.80	0.72	0.74

B1, B2, B3 - 25%, 50% and 100% labeled data in Italian

MM - Mixture Model without labeled data, **Sup** - (Small portion) Supervised labeled data

Discussion

- Engineering good tokens and lexicon is key - crazy social media vocabulary - ***coolio, ssup, AFAIK, IMHO, IDK, LMFAO, I am happyyyyy….., Typos***

- Top words in English and other languages are not exact translations.

- EM - Complexity : O(|V(source)| * |V(target)| * |Classes(2)|)

- Semi-supervision on the model - Has value and needs to be engineered better.

- **Maintenance Costs** : Train lexicon and EM once in every new language(and at times when you think your vocabulary increases).

In a nutshell....

What we really wanted : Compact and single model, knowledge transfer, scales across languages

Cross-lingual Word Embedding	Mixture Model
Knowledge Transfer(KT) through pre-training and alignment	KT through word translation (bilingual lexicon) mixtures
Similar neighbors, Classifier flexibility	Interpretability, Classifier flexibility
Single model, Compact representation	Single model, Saves data labeling drastically

Future Work

Improve words tokenization to train dynamic embedding	Train better lexicon
Improve alignment: more frequent words	Approximate Inference

Tribe Dynamics

Visualization Tool

pancake

Word-Embedding visual model

Visualization Tool

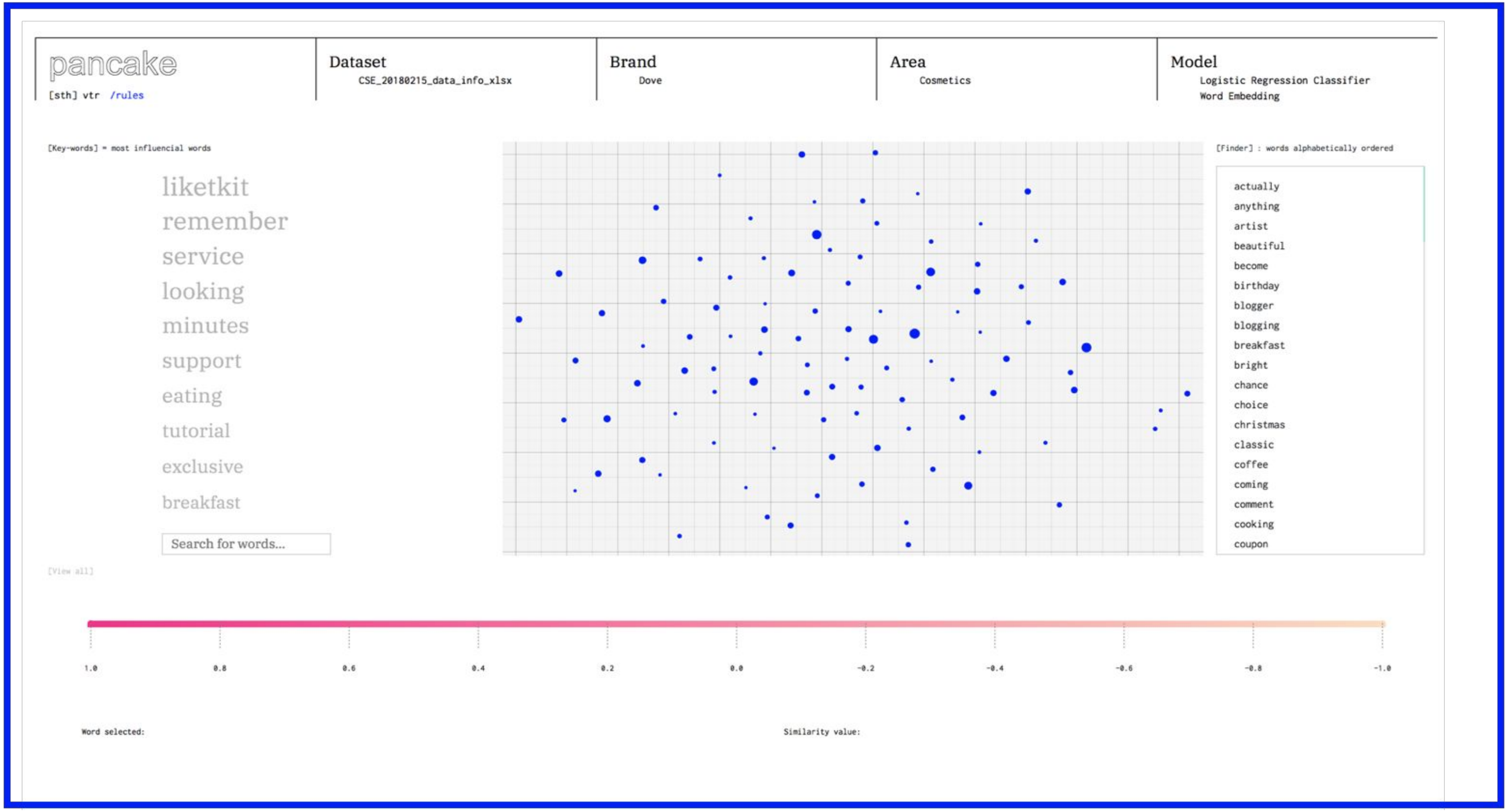
Users: Model Developers

Scope: Help the developers to check the model’s result

What the tool does: print the words by their similarity

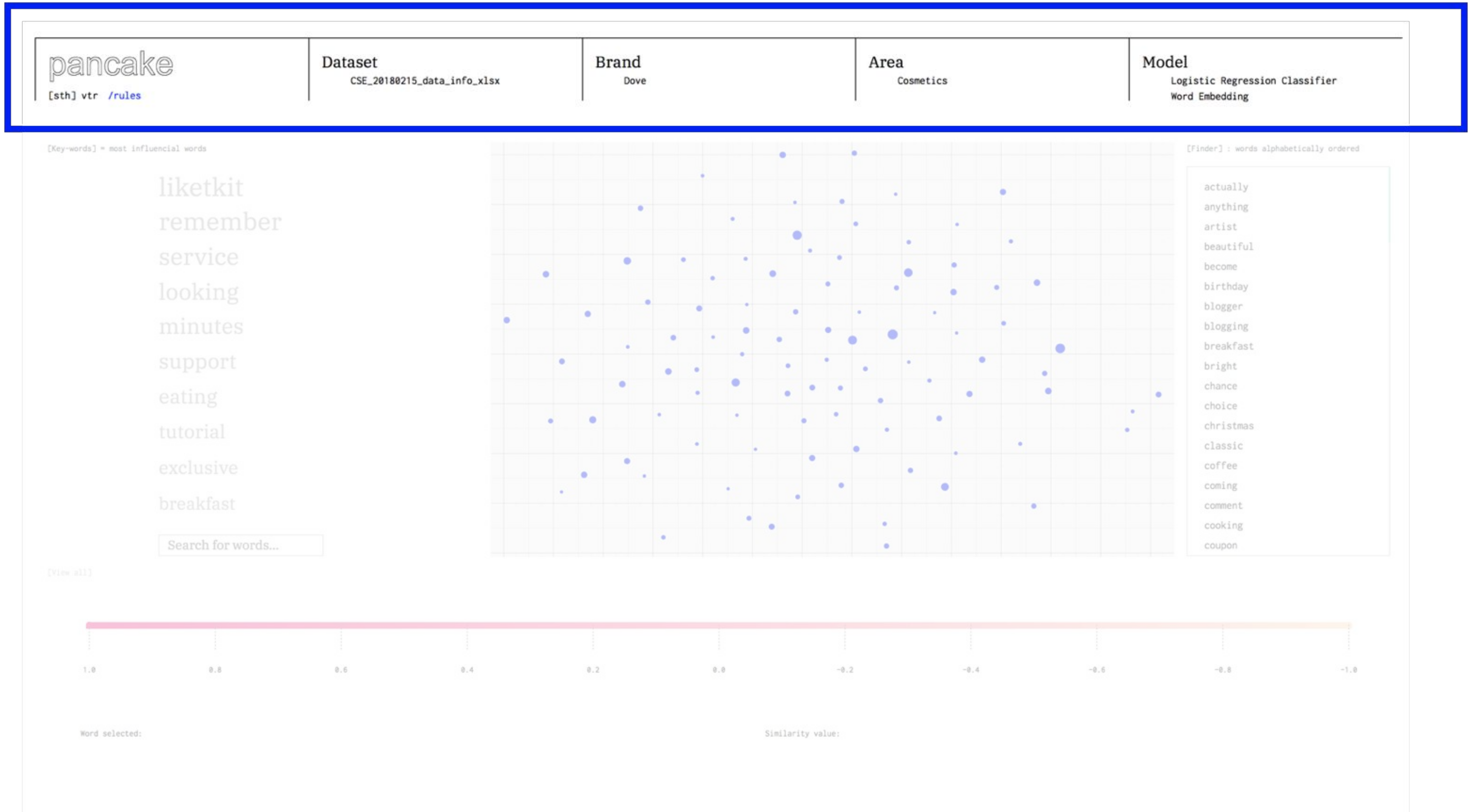
Tool Link: https://datashack.github.io/tribedynamics/Visualization/index.html

Landing page



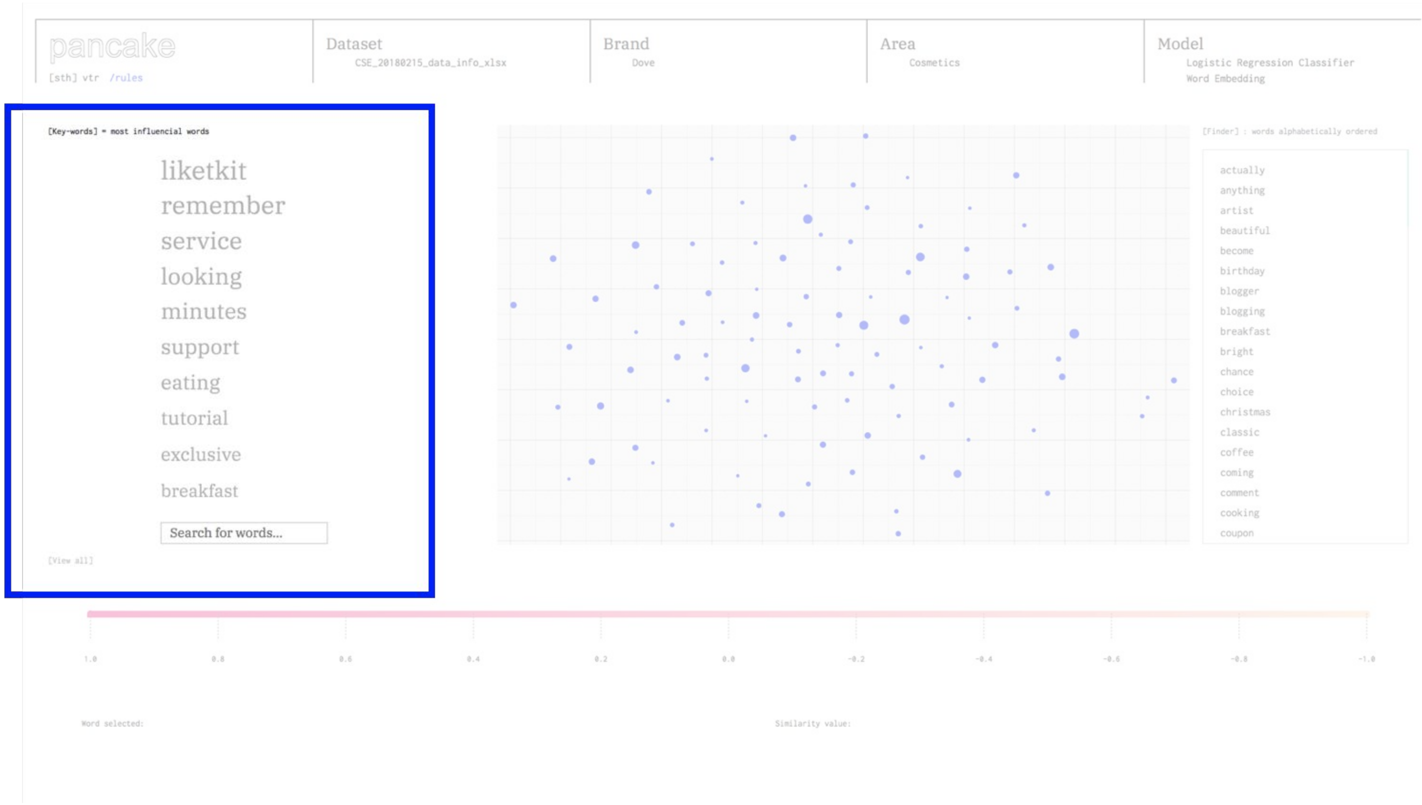
Header

To remind to the developer on which dataset he is is working on and the models that have been applied



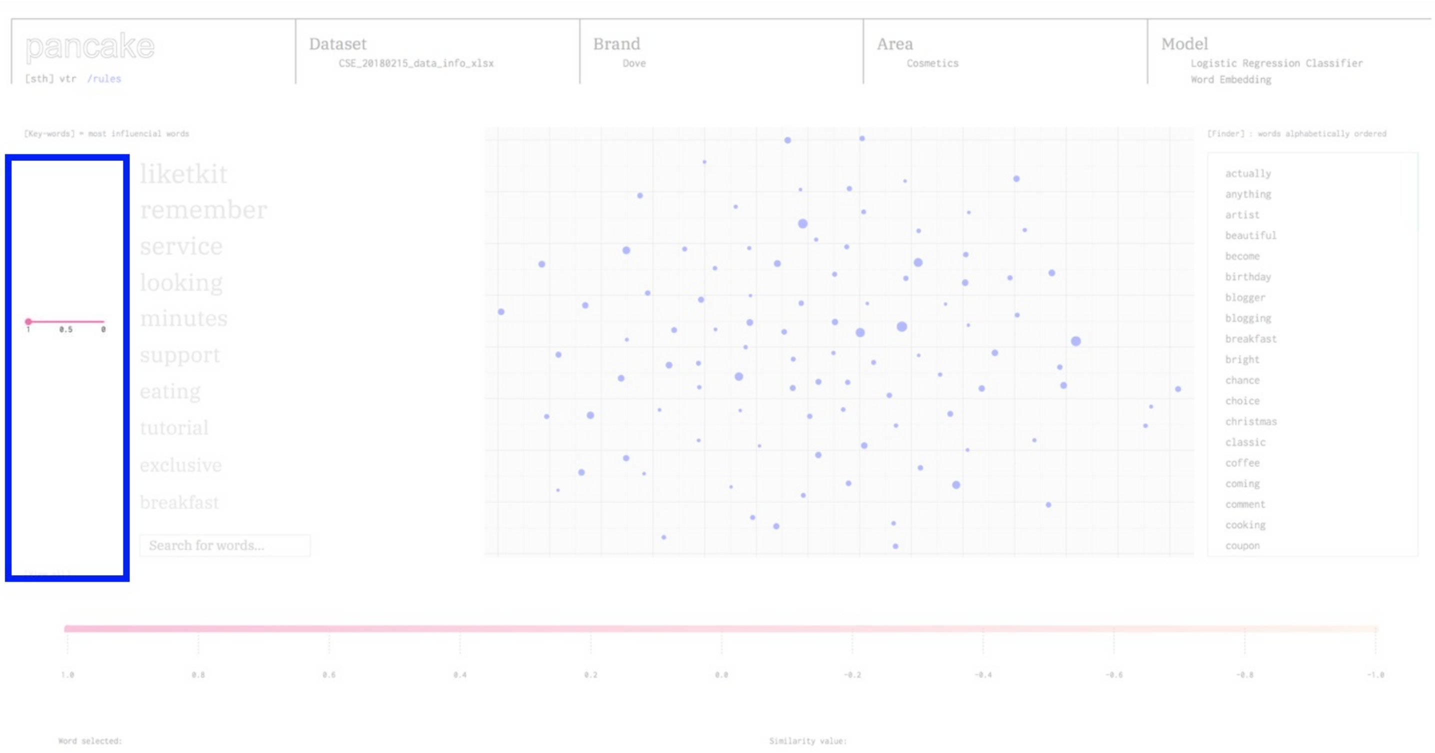
Key-word

List of the most influential words for the text classification task, according to a Logistic Regression Classifier



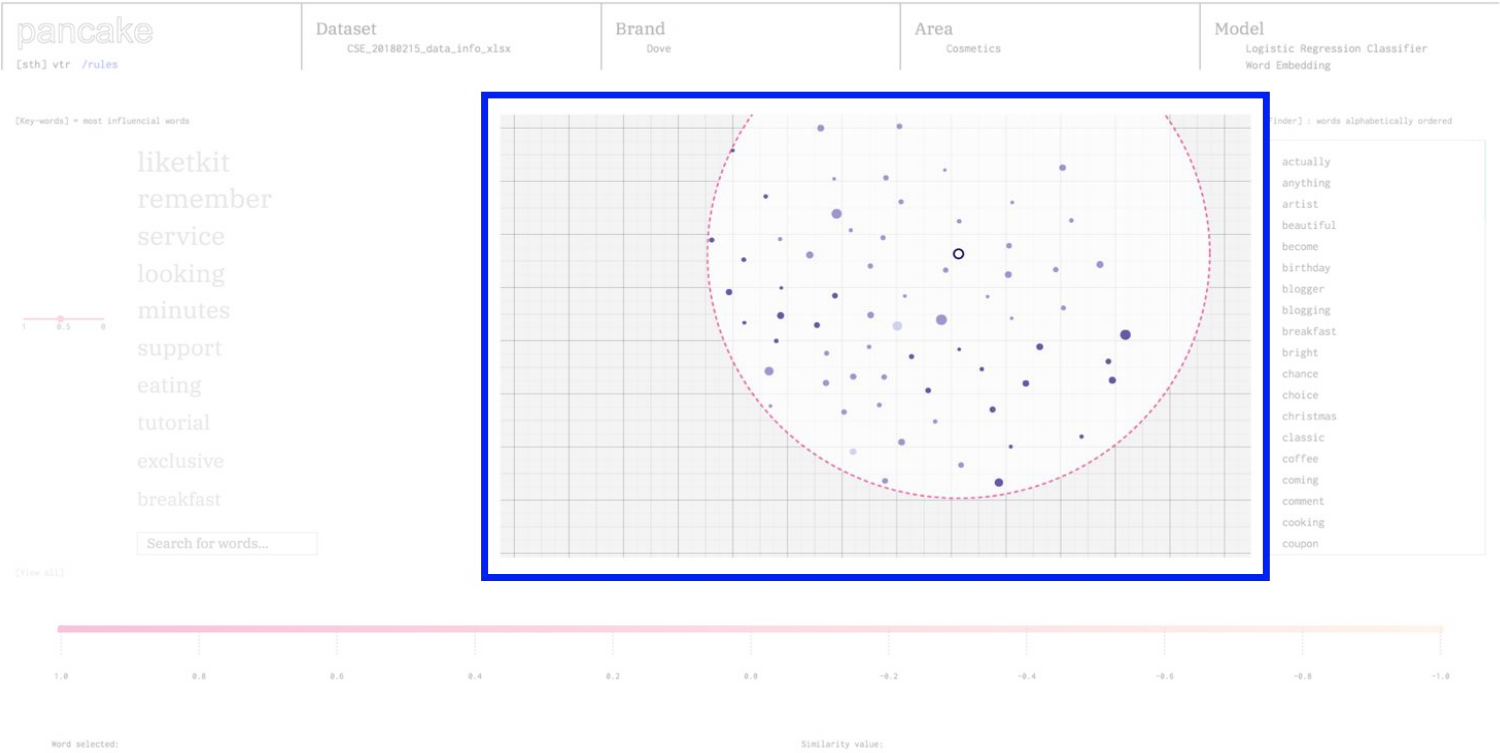
Slide bar

The user can set the similarity range to select the words to analyze



Scatter-plot

Words are positioned in two dimension after having been reduced from 300 to 2 dimensions using TSNE dimensionality reduction



Scatter-plot

How the scatter-plot is built

Dots color

Delta small → Delta large

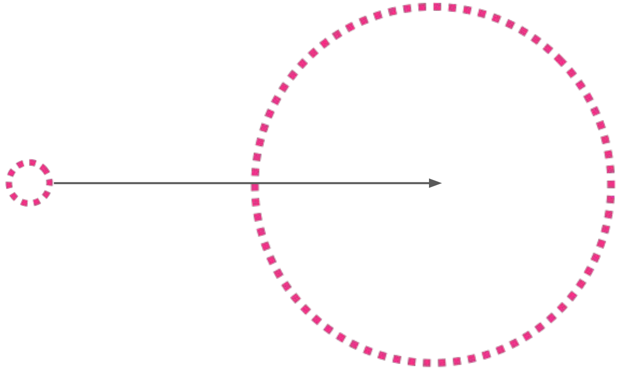
Formula to define dots color:

color_i = delta_i = |cosine_similarity(selected word in 300 dimensions, word_i in 300 dimensions) - cosine_similarity(selected word in 2 dimensions, word_i in 2 dimensions)|

Dots dimension is depending on influence value

High influence → Low influence

Circle selector and slide bar



Small selection → Large selection

1 = [corresponding with word selected]