university of groningen

Text Categorization in Humans and Machines

Daniel Mark Low

Master's Thesis

MRes in Language and Communication Technologies,
University of Groningen
MSc in Cognitive Science, University of Trento

Advisors
Scott Laurence Fairhall
Barbara Plank

Academic Years 2016-2018

Acknowledgements

I would first like to thank Scott Fairhall for being a fantastic advisor. You first suggested this project when I was quite unqualified, but we took a chance and at the least I became much more qualified in the process, which is a nice life lesson. Thanks for all the countless hours on Slack, for helping me attend two amazing summer schools, and for supporting all my PhD applications. And thanks for giving me total freedom to be self-motivated and learn from my own mistakes.

I would also like to give a special thanks my other amazing supervisor, Barbara Plank. Thanks for all the amazing feedback and insights. I hope more researchers shared not only your big-picture views for the field but also your kindness and generosity.

Thanks to all the members of the Fairlab at CIMeC. Special thanks to Natasha for all the amazing feedback, patience and dedication. This project is also yours. Thanks to Aidas for teaching me more of the devil's programming language, for being a bro and offering me your couch. Thanks to Silvia and Elisa for giving me all your time and feedback as participants during the pilot. And for all the amazing lunches.

Thanks to Gilda in Rovereto for being the best cafeteria in the world and inspiring me to make beautiful things.

Thanks to Sushrut for all our epic conversations on machine learning, generalized AI, and life in the streets of Rovereto and Genoa. Thanks to Mostafa for teaching so much NLP during our awesome runs and lunches around Groningen and thanks for sharing the B-side of the Beatles with me.

Thank you Evelina Fedorenko and your lab for the helpful discussions with the experiment design.

Thank you so much to the Erasmus Mundus European Masters Program in Language and Communication Technologies (EM LCT) for their support throughout these two years.

Thanks to the Center for Brain, Minds and Machines and Neurohackademy for inspiring me and helping me change my views on AI and science.

Thanks to my family and friends. Inmeasurable thanks to Julia, for teaching me more than anyone and supporting me every day.

While we're at it, thanks to the open source community, especially the developers of Python, Keras, Tensorflow, Seaborn, Scikit-Learn, Wikipedia, and Dbpedia. Thanks to all your software, tutorials, books, and stackoverflow answers, a guy like me who started this project with little computational background could learn so much.

Contents

Abstract

In recent years, there have been many attempts of decoding computational representations of sentences from the associated brain activity of human's reading these sentences during functional Magnetic Resonance Imaging. However, most of these studies have not interpreted what information is encoded in the representations, while assuming that these representations capture general constructs such as "semantics". One goal of this thesis is to demonstrate how computational representations can be characterized by how much information they contain related to many linguistic and semantic properties including animacy, semantic similarity, or word order to show they encode more than just the task they were trained on (e.g., text categorization). Therefore, if certain regions decode these representations, these regions can be better described. A second goal is to show the value of using complementary computational representations of the same task for brain mapping. For instance, different models –from logistic regression trained on tf-idf features to a deep neural network trained on word embeddings– accomplish text categorization in different ways. It is then possible to see which which brain regions have information on all forms of text categorization and which brain regions are specific to each model.

We first train models on text categorization. We then show how these models can be mapped onto the brain using Representational Similarity Analysis. However, in order to understand what information is mapped to each region, we provide a series of methods to analyze representations. We first compare each model's performance with human judgments on semantic textual similarity and text categorization to understand which model best captures human processing. The human sentence categorization task is carried out on sentences with more than one plausible category, which creates a challenge for any computational model that wishes to capture human intuition. Finally, we use a series of techniques to interpret each model's representations including probing, distorting inputs and architecture, Representational Similarity Analysis, and measuring how superficial or abstract each hidden layer is. We expect models that are similar in their interpretations to produce similar brain mappings.

Keywords: semantics, categorization, compositionality, convolutional neural network, long short-term memory, deep learning, brain, fmri.

1. Introduction

"Essentially, all models are wrong, but some are useful" - George Box

"Truth is much too complicated to allow anything but approximations" - John von Neumann

Humans are constantly categorizing the world around them. We categorize explicitly when remembering that Fitzcarraldo is a Werner Herzog film or more implicitly while trying to comprehend a sentence such as "He wrote many famous songs". We know the sentence is about something animate and very likely a musician while knowing a sentence with a very similar meaning such as "It has many famous songs" is about something inanimate, an album. Powerful models from formal semantics (e.g., Montague (1970)) have described the type of knowledge and rules that humans may have in order understand how the meaning of individual words change when they are composed into phrases; for instance, to know that the meaning of "red" in "the red light" is quite different than "red" in "he's a redhead". Interpreting the exact meaning of a phrase or its overall category are types of semantic compositionality. Furthermore, semantic categorization is probably occurring in parallel when interpreting the precise meaning of a phrase.

1 Decoding Semantic Compositionality in the Brain.

In recent years, there have been many attempts of using feature vectors to predict brain patterns associated to the meaning of words (Mitchell et al., 2008; Abnar et al., 2017) and sentences (Anderson et al., 2016; Pereira et al., 2018; Anderson et al., 2018; Jain and Huth, 2018). Two complementary approaches are used: encoding, which uses stimuli (e.g., word embeddings) to predict brain activity and decoding, which uses brain activity to predict information about the stimuli (Naselaris et al., 2011).

Gauthier and Ivanova (2018) summarize a series of issues with these studies. First, these models assume that since the stimuli feature vectors encode "meaning" –which is arbitrarily and vaguely defined–, then the predicted brain regions are "semantic representations". However, computational representations not only encode "meaning". For instance, when studies decode brain activity from averaged word embeddings of a sentence (as in Pereira et al. (2018)), word embeddings encode meaning as well as different aspects of words such as lexical frequency, hypernymy (Fu et al., 2014) and also more complex syntactic features (Mikolov et al., 2013a). So even though being able to decode brain activity linked to sentence comprehension is an accomplishment for the field, these

studies do not specify the type of processing that is being carried out in these regions. Second, they show that models trained on a wide array of tasks (e.g., language modeling, natural language inference, machine translation, image caption retrieval, sentiment analysis) can partially predict brain activity associated to reading a sentence, and therefore, this brain activity carries aspects of each one of the task-specific models. Finally, they suggest the following recommendations: 1. commit to a specific mechanism and task to show that the decoded brain regions represent that specific task and not vague constructs such as "semantics". 2. subdivide the input feature space, for instance, by manually building a feature vector with different features (e.g., semantic, visual, syntactic) and evaluating the complete and partial model to see which regions belong to which features, and which features overlap in the brain. 3. Explicitly measure explained variance. For instance, Anderson et al. (2018) predicted "brain activity encodings for individual stimuli such as the mention of a specific story character, the use of a specific syntactic part-of-speech or the occurrence of a given semantic feature".

Here we wish to take this proposal a step further: not only should a specific task be specified, but we wish to demonstrate that several models trained on the same task can provide complementary and useful information. We develop a framework for mapping interpretable models of a given task onto the brain (see figure 1.1). We use four different models on the task of semantic categorization and analyze their representations to show common and different information. All models can be used to predict overlapping and different brain regions dedicated to text categorization in order to find more specific representations. We hypothesize that this task may elicit different brain regions than a paraphrasing task (which encodes something closer to the exact meaning of the sentence). Furthermore, categorization models have given promising results in the field of vision (Cichy et al., 2016), where each layer of a convolutional neural network can be mapped to the brain following the visual ventral stream which is known to gradually encode semantic abstraction. However, there remains to be studies using language to show how the language categorization model may reveal linguistic semantic abstraction (i.e., reveal areas specialized for single-word meaning onto areas specialized for interpreting the category of a sentence).

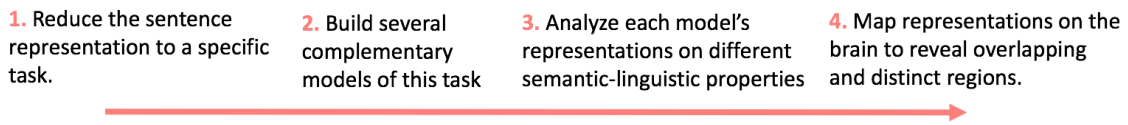


Figure 1.1: Our approach to map interpretable and complementary models of a given task on the brain.

In chapter 2, we describe the design, training, and testing of these complementary text categorization models from which feature vectors will obtained (step 1 and 2 in approach). In chapter 3, we provide the design for an fMRI experiment to predict brain regions important for different levels and types of text categorization (step 4; this is presented before interpreting the models in order to understand the goal of this thesis and interpreting computational representations). In chapter 4, we start analyzing the feature vectors by comparing their predictions to human judgment (step 3). We also provide a challenge to any computational model that wishes to encode a human intuition of text categorization: models must predict the category of sentences with more than one

possible category and compare their predictions to human judgment. This should start to answer which model best captures human intuition. In chapter 5, we apply several methods for analyzing and comparing each model's representations to know what is actually being encoded in potential brain regions (step 3).

2. Text Categorization in Machines

We chose four different text categorization models that cover a diverse array of semantic compositionality needed for text categorization: 1. logistic regression trained on tf-idf features, 2. logistic regression trained on averaged word embeddings 3. a convolutional neural network (CNN), and 4. a long short-term memory (LSTM). The first two models are simple linear models and the second two are deep neural networks. Deep neural networks allow the possibility of higher performance, a different type of compositionality, and intermediate representations of that compositionality from the hidden layers. Each one will be reviewed in how they theoretically and empirically model the compositionality of text categorization. The goal of this first experiment is to see which model has the highest performance on a new data set on the task of text categorization. Furthermore, feature vectors will be obtained from each model for further experiments.

1 Review of Models with Different Types of Compositionality

Each one of the four models will described in turn. Table 2.1 summarizes theoretical types of compositionality that we assume each model seems to be performing.

Model	Type of compositionality	Levels
LogReg Tf-idf	No compositionality: shortcuts compositionality by weighing important words per category.	No
LogReg avg. embeddings	Composition through averaging word embeddings of a sentence without stop words.	No
CNN	Hierarchical composition from local compositionality (ngrams) in first layer to global compositionality (category) in final layer.	Hidden layers
LSTM	Cumulative composition word by word.	Time steps

Table 2.1: Types of compositionality. "Levels" means the degree of compositionality either throughout hidden layers (both deep neural networks) or time steps (LSTM).

1.1 Term Frequency-Inverse Document Frequency

small edits as requestedTerm Frequency-Inverse Document Frequency (tf-idf) is a method of assigning weights to words to measure the importance of a given term in a document (Salton and Buckley, 1988). Tf-idf carries the intuition that a term that occurs in many documents (e.g., the pronoun "she") is not a good discriminator and should be given less weight than a term that occurs substantially more in a single document (e.g., "pilot" in an document on a pilot), which would mean it is representative of that document. The classical formula is:

$$w_{i,j} \ = \ tf_{i,j} \times log(\frac{N}{df_i})$$

where $w_{i,j}$ is the weight for ith term in jth document, N is the number of documents in the collection, $tf_{i,j}$ is the term frequency of ith term in jth document and df_i is the document frequency of ith term in the collection (Zhang et al., 2008).

Logistic regression. The resulting feature vectors are then trained with a logistic regression classifier. In the binary logistic regression problem, there is a binary output variable Y, and the goal is to model the conditional probability Pr(Y = 1—X = x) as a function of x (Shalizi, 2013). When applying linear regression to this probability problem, a first attempt could make p(x) be a linear function of x, where an increment in x would add or subtract a certain amount to the probability. However, p must be between 0 and 1 and linear functions are unbounded. Therefore the solution is to first apply a log transform so that changing an input variable multiplies the probability by a fixed amount, and then modifying p with a logistic transformation or logit since logarithms of probabilities are unbounded in only one direction and linear functions are not. The logistic regression model would be:

$$\log \frac{p(x)}{1-p(x)} = \exp(\beta_0 + \beta_1 + \beta_2 + \cdots + \beta_n$$

For multiclass classification, we simply compute this logit probability for each class. For a given class (e.g.,"Aircraft") and a given input (e.g., 300 dimensional word embedding), when solving for p, the formula would be:

$$\Pr(\text{Aircraft} \mid \text{300D word embedding}) = \frac{\exp(\beta_0 + \beta_1 + \beta_2 + \cdots + \beta_{300})}{1 + \exp(\beta_0 + \beta_1 + \beta_2 + \cdots + \beta_{300})}$$

It would be reasonable to assume humans have the intuition of tf-idf: a way of categorizing a text would be to evaluate which category the content words belong to the most. However, there is no composition in tf-idf since a sentence is simply a concatenation of term weights. Most weights will be zero, creating a sparse vector, which is practically a "mask" of zeros over irrelevant words. Furthermore, returning to the example in the introduction, tf-idf would probably not return a strong distinction between "She wrote many famous songs" and "It contains many famous songs", since pronouns such as "she", and verbs such as "writing" and "contains" are not particularly important for the sentences' categories.

1.2 Averaged word embeddings

Given a sentence, each word can be represented by a word embedding, a multidimensional vector (e.g, of 300 elements) that represents its meaning obtained from co-occurences in large corpora (Mikolov et al., 2013a)[1]. A very basic form of composing these units of meaning is taking the element-wise average of these word vectors, also known as bag-of-words. Due to averaging, the sentence vector does not encode word order. And it is possible to include only content words because the stop words could re-orientate the multidimensional vector in a sub-optimal and semantically meaningless direction, as only content words tend to carry meaning (Mitchell and Lapata, 2008; Mikolov et al., 2013b). These resulting feature vectors are also trained with the logistic regression model described before.

This simple approach has obtained surprisingly high performance in different types of task (Hill et al., 2016). However, from a cognitive point of view, averaging units of meaning is hard to interpret. Furthermore, removing stop words creates an artificial input space. Cognitively plausible models should learn to extract or disregard information from stop words within phrases as humans do.

1.3 Convolutional Neural Network (CNN)

CNNs had generally been used for computer vision (Krizhevsky et al., 2012) inspired by the brain's visual system (Goodfellow et al., 2016). However, recent studies using CNNs have performed extremely well on sentence classification tasks (Kim, 2014; Zhang and Wallace, 2015; Conneau et al., 2016). Many CNNs for sentence classification have a single convolutional layer. However, a CNN more hidden layers enable analyzing how the semantic representation changes at each layer, which is of particular interest for us since we can find different levels of representation throughout the brain.

In figure 2.1 we describe a the multi-layer CNN architecture for text categorization. We added a dense layer between the last maxpool layer and the softmax layer to create a bottleneck from the output of the maxpool2 to the last layer. The dense layer allows the model to reorder what it has learned up until the second maxpool layer (which still maintains ngram order) if necessary to improve performance.

A model with at least three layers is preferable to obtain a gradual composition of meaning from sentence to category –from more concrete and local composition of a few words in layer 1 to more abstract and global in the last layer. If the brain has degrees of abstraction, one could assume that at some point there is a single representation for the whole sentence when it is read and a dense layer seems to better characterize this type of representation.

ELU is preferred over ReLU because the latter creates sparse feature vectors since it flattens negative values to zero. Since our goal is to correlate the feature vectors from the hidden layers of the CNN between sentences for Representational Similarity Analysis (see Chapter 3), a more nuanced feature vector (with different positive and negative values) is preferred (however, we include ELU vs. ReLU comparisons in the gridsearch analysis and choose the highest performing after the dense layer).

[1] See online tutorial for introduction to Natural Language Processing: https://github.com/danielmlow/nlp_tutorial/blob/master/tutorial.ipynb

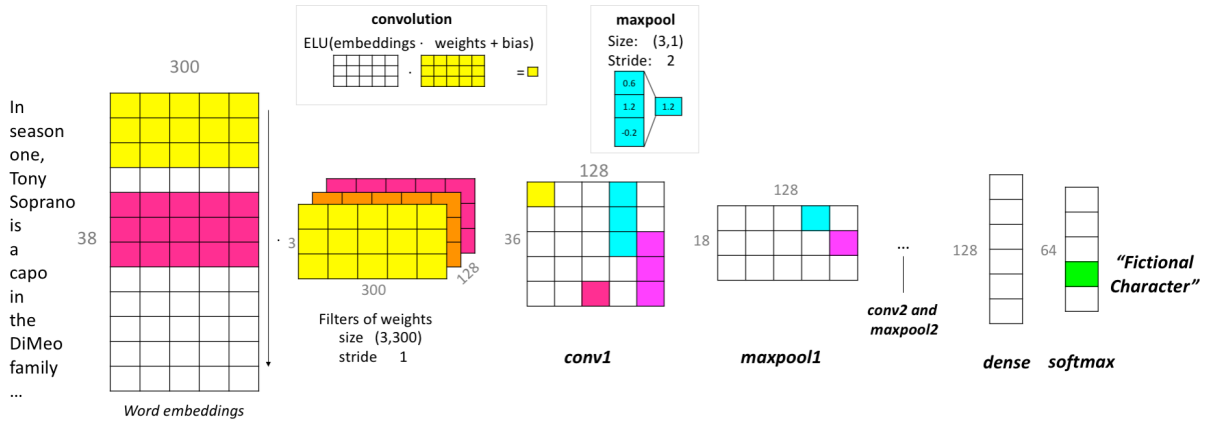


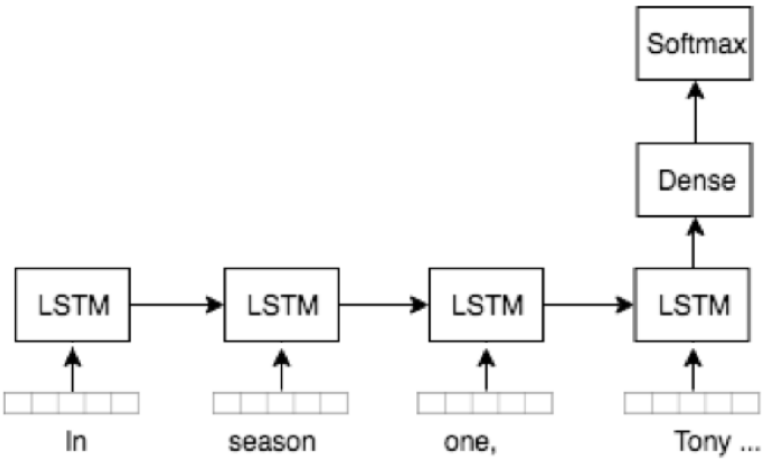
Figure 2.1: Convolutional neural network architecture. The input is a sentence matrix, which is built by concatenating the word embeddings of each word (of size 300 in this example). If using sentences up to size 36 and a filter of weights of size of three, three rows of the sentence matrix are repeatedly multiplied (through a dot product) by a single filter of weights of size 3x300, shifting one row at a time (i.e., a convolution). Each filter contains weights that are randomly intiliazed and later learned through training. Applying the filter to the input through a dot product ouputs convolutional layer 1 (conv1) of size 36x1. Then a bias term and an activation function are applied to each feature vector. The bias is learned to determine how important that conv1 neuron is for the network. If the bias is low (negative) than the input and weight must be high for the neuron to be active; if the bias is high (positive) than even small inputs will make it activate. This $activation function(inputs \cdot weights + bias)$ computation is done with multiple independent weight filters (128 in the example) to learn complementary information of the same input. Maxpool is then performed to obtain one value from each feature vector resulting in maxpool layer 1. Maxpool seeks to downsample the information, keeping only the highest features. This process can be repeated with multiple conv-maxpool layers. Finally, there is a fully-connected dense layer which results in an ouput layer of 64 elements.

1.4 Long Short-Term Memory (LSTM)

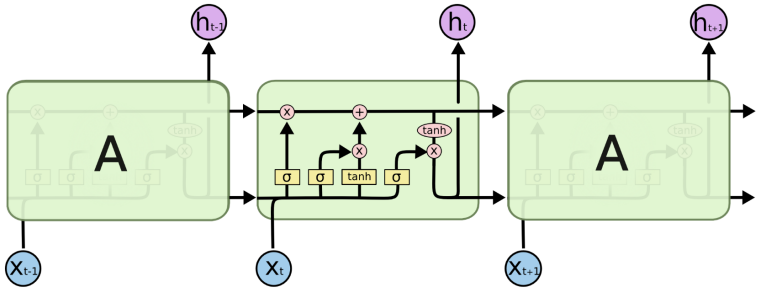
Recurrent neural networks (RNN) were introduced to model sequential and dependent inputs. They are recurrent because they apply the same process for every time step of the sequence with the output being dependent on the previous time steps. The issue is that these models carry a "memory" of only a few previous time steps. RNNs also have high time complexity and can end exploding gradients or vanishing gradients, which make time complexity depend exponentially on the size of the weights:

Exploding gradient. An error gradient with respect to the weights is the direction and magnitude calculated during the training of a neural network that is used to update the network weights in the right direction and by the right amount. Exploding gradients occur when weights with values larger than 1.0 are repeatedly multiplied (Pascanu et al., 2013). This makes the network unstable during learning, which when the loss function has a high oscillation, instead of an optimal steady decrease.

Vanishing gradients. Certain activation functions (e.g., sigmoid and tanh) map large regions of the input space to an extremely small range. In these regions of the input space, even a large change in the input will produce a small change in the output



(a) Full LSTM architecture



(b) Content of LSTM cells

Figure 2.2: LSTM architecture

(i.e., the gradient is small). This is worse in a deep neural network: a first layer will map a large input region to a smaller output region, which will be mapped to an even smaller region by the second layer and so on. Therefore, even a large change in the parameters of the first layer does not change the output much. This means that the neurons in the earlier layers cannot learn as much and the first layers are important because they can detect patterns directly from the input.

For these reasons, the LSTM, a more efficient recurrent network, was introduced (Hochreiter and Schmidhuber, 1997). It enforces constant (instead of exploding or vanishing) error flow by truncating the gradient at certain points. Several gate units learn to open and close access to this constant error flow. Although this constant flow is not perfectly obtained, the model does learn longer term dependencies.

More formally, following the architecture in figure 2.2[2], at a given timestep h_t, an LSTM must decide whether to keep or remove an input x_t from the error flow at a given time step or cell state. It makes this decision through a forget gate which is a sigmoid layer: the network looks at h_{t-1} and the input x_t and outputs a value between 0 and 1, which is a confidence estimate for each number in the cell state C_{t1}:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Then the LSTM decides what to save in the cell state. First, a second sigmoid layer called "input gate layer" chooses which value to update:

[2] Figure (b) from colah.github.io/posts/2015-08-Understanding-LSTMs

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Then a tanh layer outputs a vector of new candidate values, C, to be added to the state:

$$C_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_i)$$

The old state is updated:

$$C_t = f_t * C_{t-1} + i_t * C_t$$

Then a sigmoid layer filters the outputs o and a tanh layer is applied to return values between -1 and 1, which are multiplied by the output of the sigmoid gate as a filter:

$$o_t = \sigma(W_o[h_{t-1}], x_t] + b_o)$$

$$h_t = o_t * tanh(Ct)$$

Deep neural networks have the advantage of having multiple levels. In an LSTM, this is true in two ways: if LSTM layers are stacked one on top of the other, the representation from the last time-step (i.e., that represents the whole sentence) can be obtained from the stacked layers to potentially achieve gradual levels of composition. A second approach is to obtain representations at multiple time-steps and therefore compare how the representation changes as the model gradually covers the input. For the latter, a single layer model seems to be better as the first time-steps will not be affected as much by a vanishing gradient then in a stacked model.

Please refer to Future Work in General Discussion for alternative computational approaches to compositionality.

2 Experiment 1: Wikipedia-DBpedia Text Categorization

2.1 Dataset

A dataset was built by linking wikipedia sentences to DBpedia classes or categories as follows. The DBpedia project (Lehmann et al., 2015) links each one of the Wikipedia articles (1.4 million in Italian) to an ontological category within an ontology or semantic hierarchy. Therefore, a dataset was built by tokenizing the sentences of every Italian Wikipedia article (i.e. the training samples), and linking each sentence to the articles DBpedia category as a training label.

Since some categories have very few articles, 64 categories were chosen from the 320 categories available to obtain similar sample sizes of 6000 sentences per category (see Experiment 1 Sentence Examples in Appendix for categories and sentence examples). We wanted to maximize the amount of categories covered. 6000 samples were selected because lowering it to, for example, 5000 would only provide three more categories and increasing it to, for example, 12000 samples would reduce the semantic space considerably to 54 categories that have that amount of sentences.

Sentences are between 6 and 38 words, since the goal of this study is to present sentences to humans and sentences over 40 words are more likely to change topics. Furthermore, median Wikipedia sentence length was 20 words and therefore we left out sentences less than 6 words (percentile 4) and more than 38 words (percentile 89), leaving out 15% of the sentences.

An 80-20 train-test split was applied which resulted in a training set of 4800 sentences per category (used for grid search and cross validation) and a test set of 1200 sentences per category.

Other datasets were considered: 1. Using only the abstract of the artcile instead of the whole article. However, was not preferred since it creates less sentences per categories which either hurts the sample size or the amount of classes (i.e., semantic space); a less fuzzy semantic space which is not preferred since humans seem to have a fuzzy semantic system (Binder et al. (2016)): semantic categories are not divided in the brain but overlap and connect in many different ways, so the type of fuzzyness captured by Wikipedia articles is optimal. 2. The DBpedia label of the article may not apply to certain sentences within the article (e.g., an article on the humming bird will describe not only birds, but also its discoverer or its ethymology). Therefore, unsupervised topic modeling of sentences was sought out but resulted in many redundant topics over the first 15 topics, resulting in a small semantic space.

2.2 Models

Logistic Regression with Tf-idf. The default scikit-learn Logistic regression parameters[3] were used including one-vs-rest (OvR) scheme. Tf-idf was used with scikit-learn default parameters[4] except with a minimum document frequency $= 2$ and an ngram range from 1 to 3 words to match input information of the CNN.

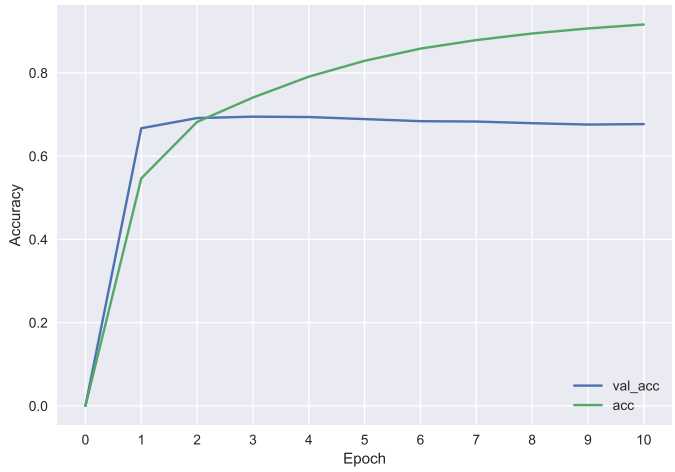
Logistic Regression with word embeddings The same logistic regression model was used as with the tf-idf model. We used Google word2vec Italian word embeddings. We left only content words because stop words tend to not carry meaning but nevertheless carry the same weight in reorienting the sentence vector. (Bojanowski et al., 2016). It possible to choose the highest performing option, but since this work cares about the interpretability of the representations, the reasoning is the averaged sentence vector $mean(the, airline, flies, to, Africa)$ would be in a worse semantic direction for the category Airline than $mean(airline, flies, Africa)$ (**?**).

CNN. Once we designed a basic architecture as described in the previous CNN section (see figure 2.1), we optimized the hyperparameters for accuracy. To reduce the amount of hyperparameter comparisons, we split the grid search in three steps: 1. We first found a reasonable epoch size by starting the model on intermediate or standard values for the hyperparameters, and ran it for 10 epochs with the following parameters: drop: 0.3, batch size=256, optimizer: Adam, activation1: elu activation2: elu. The validation accuracy was the highest after three epochs (see figure 2.3 for learning curves).

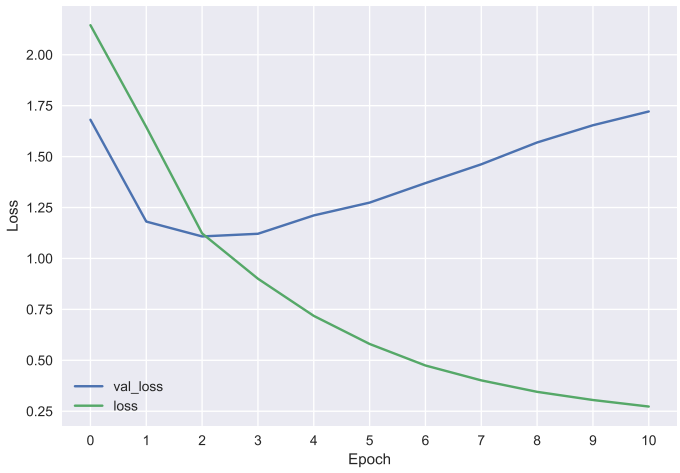
2. We than ran a grid search on hyperparameters pertaining to key architecture design: number of filters: 32, 64, 128; amount of neurons in first dense layer: 1024, 512, 256; amount of neurons in second dense layer: 64, 128; best parameters were:

[3] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html



(a) Accuracy



(b) Loss

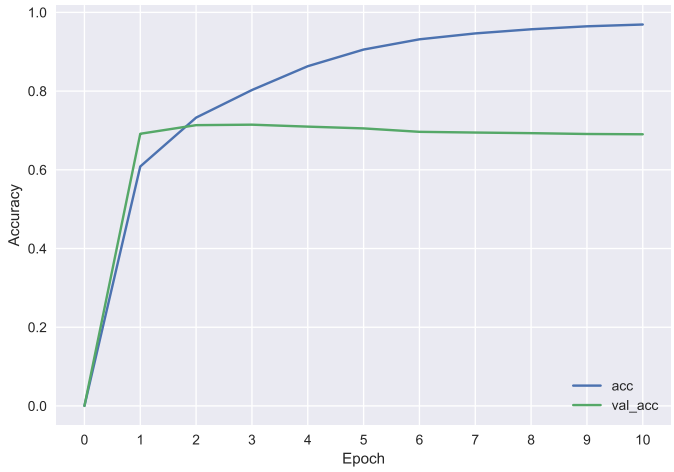
Figure 2.3: CNN accuracy and loss learning curves

number of filters: 128; dense 1 neurons: 512; dense final neurons: 128; activation in convolutional layers: elu.

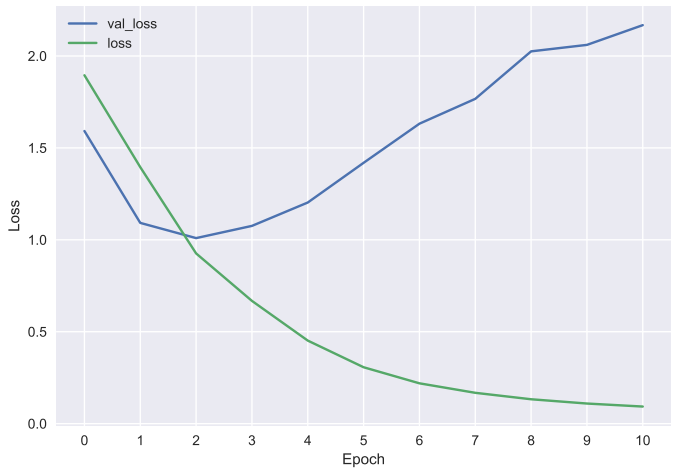
3. We ran a grid search on the remaining key parameters: dropout rates: 0.1, 0.2, 0.3, 0.4, 0.5; batch sizes: 64, 128, 256, 512; activations in dense layers: elu, relu.

Final parameters after tuning:

- dropout rates: 0.2;
- batch size: 512;
- optimizer: Adam;
- activation conv layers: elu;
- activation dense layers: elu;
- number of filters: 128;



(a) Accuracy



(b) Loss

Figure 2.4: LSTM accuracy and loss learning curves

- dense final neurons: 128;

This procedure is suboptimal since there could be possible interactions between the hyperparameters of each step. Furthermore, there alternatives to gridsearch such as random search (Bergstra and Bengio, 2012). However, this is a proof-of-principle of how models can be interpreted for their linguistic-semantic properties and mapped onto the brain.

LSTM. The architecture consisted of a single LSTM layer with a dense layer and softmax layer (see figure 2.2 (a); see table 5.2 for results on an LSTM with two stacked layers). A single LSTM was preferred since the compositionality of interest was that across time-steps, not layers, as we are trying to build models with different types of compositionality. As with the CNN, an initial model found three epochs to be optimal (see figure 2.4 for learning curve).

Then grid search was performed with the following parameters: Dense final neurons: 512, 128; dropout rates: 0.1, 0.2, 0.3, 0.4, 0.5; batch size: 128, 256, 512; optimizer: Adam, activation for LSTM layer: elu, relu; lstm neurons: 300, 600.

Final parameters after tuning:

- dropout rates: 0.3;
- batch size: 512;
- lstm neurons: 600;
- optimizer: Adam;
- activation: elu;
- dense final neurons: 128.

Since each sentence can have more than one reasonable label, top 1, 2, 3, and 5 accuracy will be computed for each model along with their 5-fold cross-validation (CV) score to observe the difference between validation and testing accuracy and accuracy variance within CV folds.

Deep neural networks were run with Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2015) backend on the GPU Peregrine HPC cluster of the University of Groningen. Logistic regression was run with sklearn (Pedregosa et al., 2011).

2.3 Results & Discussion

Table 7.2 shows the full results. The standard deviation in cross validation is low and therefore the data set is relatively consistent across folds. The difference between the cross-validation scores and the test score is also quite low; therefore, results from validation generalized well to the test set.

The highest performing model was the LSTM (for the LSTM classification report, see section 1.1.Classification Report in Appendix). The accuracy increases substantially from top-1 to top-5 accuracy, reconfirming that this dataset has fuzzy labels. It is reasonable to assume that there are several appropriate labels for most sentences. Therefore, the top-3 or top-5 accuracy seem to be more valid estimates of performance.

Since the goal is to compare models, one interesting metric is Cohen's Kappa score for inter-rater agreement. Each model's predictions for test set were compared as raters, which resulted in kappa $= 0.73$ (and almost identical scores for other metrics such as fleiss). This agreement score is considered substantially high. (Viera et al., 2005).

Model	CV (SD)	Top-1	Top-2	Top-3	Top-5
Tf-idf	65.48 (.18)	66.19	77.33	82.56	87.82
Avg. w2v	67.06 (.24)	67.13	79.47	84.87	90.12
CNN	68.65 (.31)	69.52	80.85	86.00	91.17
LSTM	**71.47** (.34)	**72.32**	**83.52**	**88.24**	**92.73**

Table 2.2: Accuracy (%) on 64-way text categorization. Approximately equivalent values for f1-score. CV: Mean 5-fold cross-validation accuracy (SD: standard deviation). Top 1,2,3 and 5 accuracy on test set.

3. Understanding the Brain with Machines

In this chapter, we review an approach to map computational representations onto the brain using fMRI. As detailed in the Introduction, brain decoding studies have generally not framed their interpretations on the type of task their models were trained on (if any at all as some use averaged word embeddings). Furthermore, since it is reasonable to assume that different circuits in the brain care about different types of semantic compositionality (e.g., local, global, exact meaning, overall category), our goal is to show how the models from the previous chapter trained specifically on text categorization can be essentially mapped onto the brain and how these mappings represent different types and levels of compositionality. This approach should be revealing in that it is possible to see how the different models with different types of compositionality map to different regions as well as seeing which regions are common to all.

In this chapter, we focus on the CNN from the previous chapter to show how the final layer as well as the intermediate layers can be mapped onto the brain. Our hypothesis is that conv1 should map to areas where simple meaning representations are thought to be stored such as the inferior and middle temporal cortex (Hagoort, 2013). Whereas, final dense layer should map to areas involved in more complex and abstract compositionality, possibly typical language networks as the left inferior frontal gyrus as well as those overlapping with the default mode network Mineroff et al. (2017); Hagoort (2013). An alternative hypothesis is that layer 3 is decoded by subregions that layer 1 decoded; layer 1 may be representing information of conceptual primitives shared by many phrases, whereas layer 3 may be narrowing the representation by composing it into a specific category and therefore using less of the same regions.

1 Design of an fMRI experiment to map computational models on the brain

1.1 Create a model

Here we use the CNN from the previous chapter as an example. Any of the previous models can be used but we choose the CNN as it has been used in other studies (e.g., Cichy et al. (2016)) and particularly illustrates that each layer can be mapped. This is interesting because each layer seems to have different levels of compositionality, from local superficial representations to global and abstract representations (we show evidence for the superficial-abstract change in section 4 of Chapter 5).

1.2 Design stimuli

We need to choose stimuli that are representative of each layer while being as different as possible from other layers in order to obtain a single in fMRI data. The following approach achieves this while removing us from the stimuli selection to avoid bias:

1. Filter sentences with psycholinguistic cofounds such as mean lexical frequency, which we left within a 5 and 95 percentile-range (using the frequency dictionary from Crepaldi D (2015)) and sentence length, which was set to 14 and 15 words per sentence. This range is an adequate length for reading. We restricted sentences to a small sentence-length range because different sentence lengths carry different amount of "zero rows" due to zero-padding in the embedding input layer (i.e., sentences shorter than the maximum input length of 38 words carry the learned weights and biases in conv1 associated to zero rows in the embedding layer). By choosing sentences of 14 and 15 words with the parameters we chose and tuned, the first 14 rows out of 36 (which corresponds to the first 1792 elements of 4608 flattened feature vector) carry information of the first 16 words. The last row will carry information about words 14-16 and therefore, if a sentence has 14 words this last row will not be a zero vector. The remaining zero-rows (17-26) can be discarded from analysis. In conv2 the non-zero rows go from 1-5 (which corresponds to the first 1280 elements of 4352 flattened feature vector).

2. Using the feature vectors from the final dense layer, performed unsupervised hierarchical agglomerative clustering with a Ward criterion and Euclidean distance[1].

3. Within each cluster, choose the top 20 silhouette scores to obtain the most representative sentences from each cluster. Silhouette scores near $+1$ means the sentence is far away from the neighboring clusters. A score of 0 indicates that the sentence is near the decision boundary between two neighboring clusters and negative values indicate that those sentences may have been assigned to the incorrect cluster.

4. Recluster these 20 sentences per cluster into 64 clusters.

5. Build a RSM for the remaining sentences, but using their conv1 representations. Then for each sentence, take the mean within-cluster correlation (Spearman's rho) and subtract the mean across-cluster correlation, which enables ranking how similar sentences are to other categories. Select the 6 sentences per cluster that have the lowest scores.

6. Recluster these 6 sentences per cluster into 64 clusters. This results in uneven clusters (from 2 to 17 sentences per cluster).

7. Repeat process starting with conv1 representations and using final dense representations for the final selection.

With this algorithm, we select sentences that are representative of each layer and are not similar to the other layer. However, some are easier to understand than others out of context, even though Wikipedia offers well-curated sentences. Therefore, we ran

[1] Seaborn clustermap is used: https://seaborn.pydata.org/generated/seaborn.clustermap.html

a survey with 4 Native Italian speakers that rated the sentences in random order from 1 to 5, where 1 was very easy to understand and was very 5 difficult to understand. Then there individual scores were normalized by subtracting the mean and 4 scores were averaged. From each cluster, only half of the most easiest to understand were selected for the stimuli resulting in 192 sentences representative of each layer (384 in total). Participants gave their informed consent and were paid for their participation.

Although this stimuli set allows us to optimally compare layers, it also allows us to compare the layers with the representations from models that are not deep neural networks (i.e., tf-idf, averaged word embeddings), where this procedure is not necessary. In Chapter 4 and 5 we will analyze each model's representations to show the differences and similarities between the models.

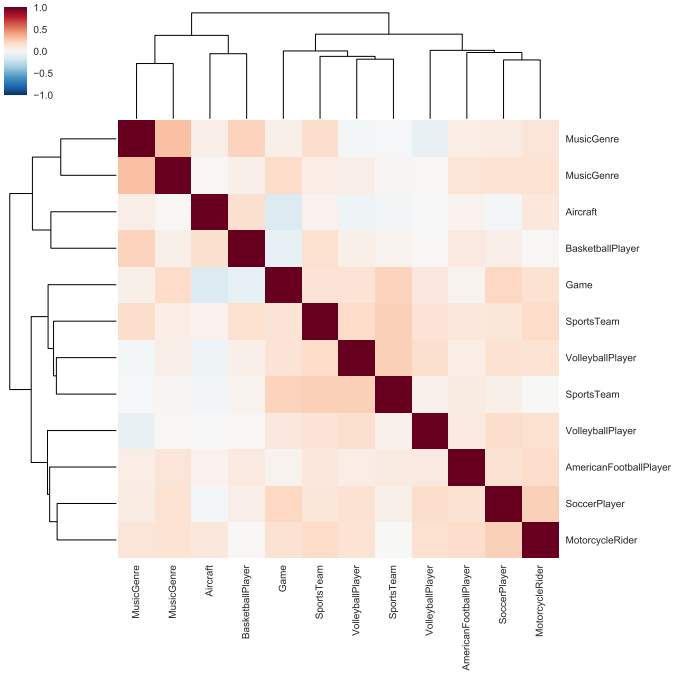
1.3 Representational Similarity Analysis

To map semantic categorization onto the brain we first use a method called Representational Similarity Analysis (Kriegeskorte et al., 2008): take the correlation (or other distance metric) between sentence's feature vectors to obtain a measure of similarity (e.g., rho) or disimilarity (e.g., 1-rho) between sentences for a given model's feature vectors. When done with N sentences, an NxN Respresentational Similarity Matrix (RSM) is obtained for each model. See figure 3.1 for the RSM of each CNN layer for a subset of 12 sentences. Then see figure 3.2 for an RSM computed by correlating the previous CNN laye RSMs.

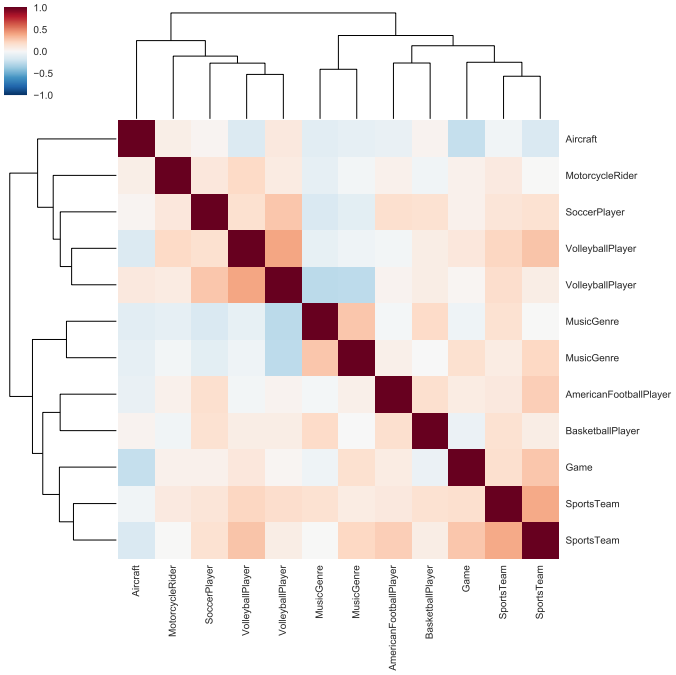
This method allows us to correlate two RSMs that carry any type of information: an RSM from a CNN layer with an RSM from brain responses (our ultimate goal described in figure 3.4), RSMs from different computational models (e.g., see figure 3.3), or RSMs built directly from human similarity judgments between pairs of sentences (e.g., see 4.5b from Experiment 3 in Chapter 4).

1.4 Searchlight analysis.

To compare models with brain responses we will use searchlight analysis (Etzel et al., 2013), which is an multivariate pattern analysis (MVPA) method (see figure 3.4). A sample of human participants read sentences from the stimuli set while undergoing functional magnetic resonance imaging (fMRI). The 3D brain volume is divided into spheres or searchlights. The searchlights are centered on each voxel and have a radius of 9 milimeters around the voxel. Reading a sentence will produce a specific BOLD activation pattern within each searchlight (i.e., a brain feature vector). Then an RSM will be built by computing the pairwise correlation between the pattern produced by reading each sentence to understand where and how the brain encodes meaning. Finally, mapping is produced by correlating model and brain RSMs. Statistically significant correlations will be regions that share information with the given CNN layer. And this process can be repeated with the different text categorization models to view where they overlap and differ in the brain.

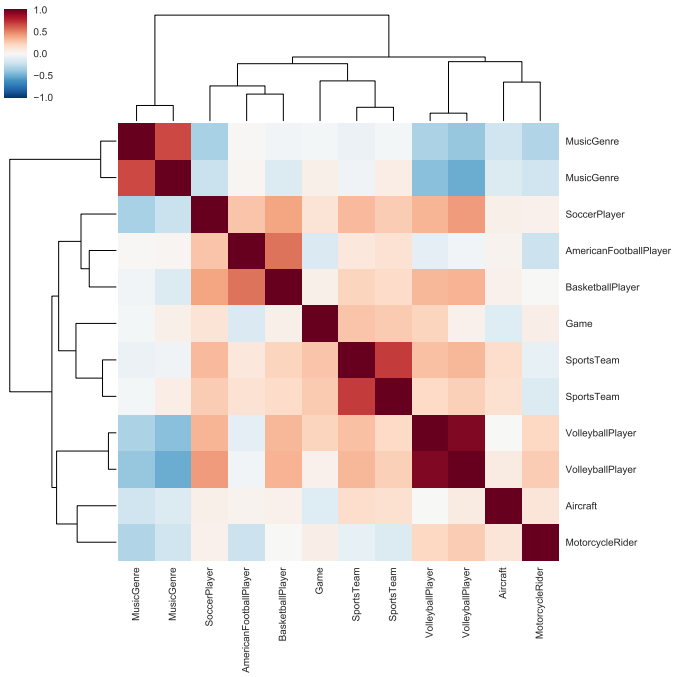


(a) Convolutional layer 1



(b) Convolutional layer 2

Figure 3.1: RSM of three CNN layers.



(c) Final dense layer

Figure 3.1: RSM of three CNN layers. Each cell is the Spearman correlation between two sentence feature vectors at the given layer. Then Ward clustering is applied to show how similarity between sentences changes from layer to layer.

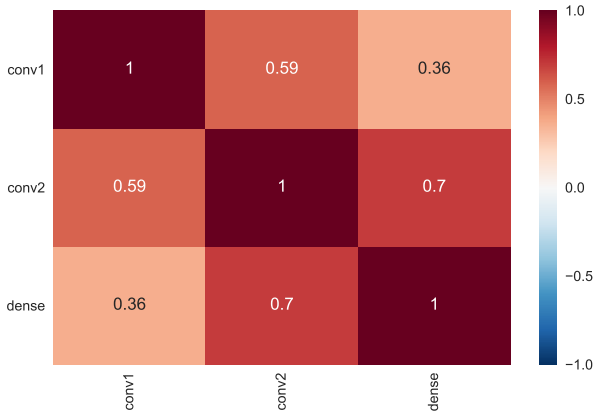


Figure 3.2: RSM of convolutional neural network layer RSMs. Each cell is the result of taking the Spearman correlation between the RSMs from figures in 3.1. Only the vectorized upper triangles without the diagonals are correlated.

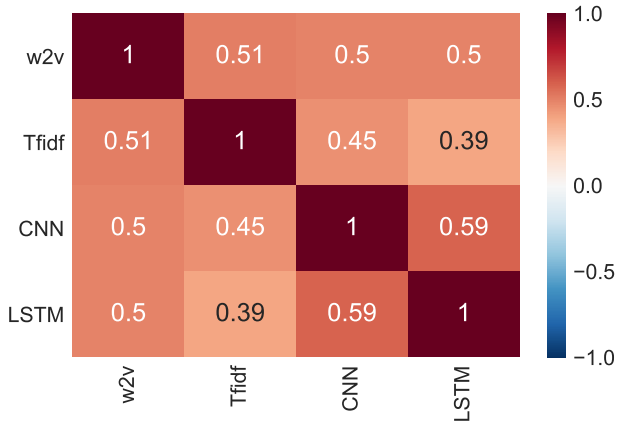


Figure 3.3: RSM of models' RSMs. First an RSM is built for the ouput layer of each model (which would be figure 3.1 (c) for the CNN. Here each cell is the result of taking the Spearman correlation between the RSMs of each model. Only the vectorized upper triangles without the diagonals are correlated.

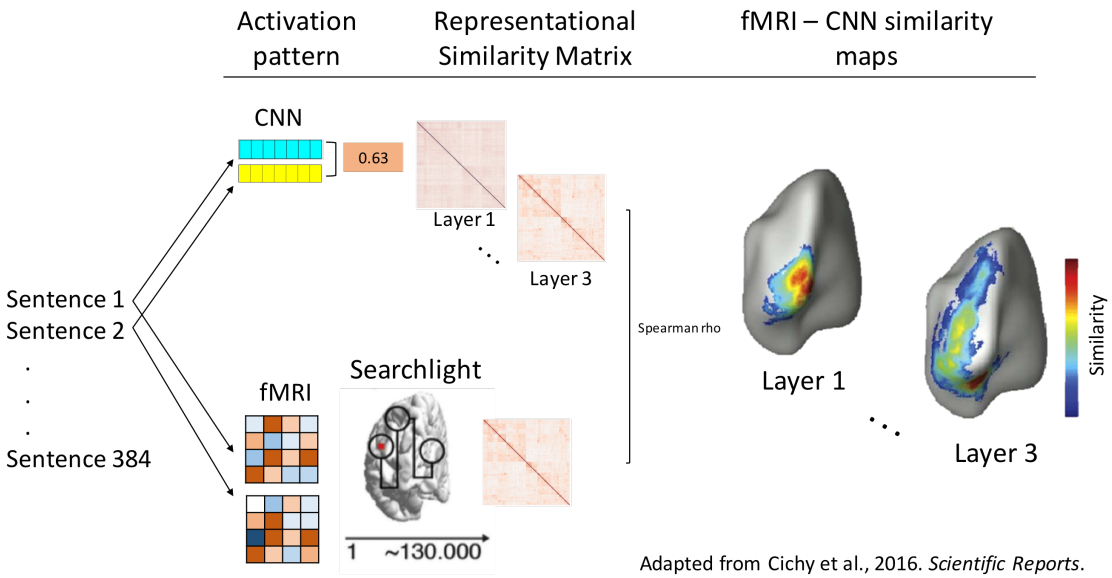


Figure 3.4: Adaptation of the description of searchlight analysis from Cichy et al. (2016). Pairs of sentences are evaluated by the CNN and read by humans undergoing fMRI. An RSM is built by computing the pairwise correlation (rho) between these feature vectors within each modality: RSMs are built for each layer in the CNN and for each searchlight on the human cortex. Then brain and model RSMs are correlated and consistent correlations mean that the model and searchlight share information. The brain mappings are from Cichy et al. (2016).

4. Understanding Machines with Humans

In the previous chapter, we showed how computational representations of sentences and different types of compositionality can be mapped onto the brain. However, what exactly is represented in each model is unclear. In this chapter, we will use human judgments to inform differences between models, that is to say, to understand which is most similar to human judgment. We will include human judgments on text similarity as well as text categorization and compare them to the models. The human categorization task is included to create a challenge to computational models of text categorization by using sentences that hard for models to categorize but categorizable by humans.

1 Experiment 2: Human Similarity Judgments

1.1 Introduction

As described in the previous chapter, our goal is to build RSMs of the different models and hidden layers to map them onto the human brain. In this first experiment, we will compare the RSMs from the model's from Experiment 1 in Chapter 1 with an RSM built from human similarity judgments of sentences. This will allow us to determine which model encodes semantic similarity between sentence's categories closest to humans. This is a natural language processing task known as semantic textual similarity. This task trains models to measure the degree to which the underlying semantics of two segments of text are equivalent to each other or are paraphrases (Agirre et al., 2016). Therefore, another way of measuring this feature is to correlate RSM of models trained on both tasks to see how much information a text categorization RSM shares with a semantic textual similarity (for more on this, see section Representational Similarity Analysis in Chapter 5).

1.2 Methods

Subjects. 26 native Italian participants were recruited through the Figure Eight platform. We requested at least 4 independent scores per stimuli and Figure Eight has designed algorithms to dynamically recruit more subjects for stimuli with high response variance (mean amount of judgments per worker = 36.3, SD = 8.3). This intends to provide a more accurate sample of human judgment. If one wished to calculate entropy or variance per stimuli, one could look at the first four judgments per stimuli. Participants were paid for their participation.

Stimuli. Three sentences from 6 categories that were clustered together in the LSTM model were chosen in order to obtain very nuanced similarity ratings (see 5.1 to see clustering between categories). The 6 categories are: Decoration, Military Conflict, Military Person, Monarch, Politician, and University. See section 3. Experiment 2 in Appendix for full stimuli set.

Procedure. Participants where shown two sentences and asked: "From 1 to 6, are these two sentences about the same topic?" (original in Italian: "Da 1 a 6, quanto riguardano lo stesso argomento queste due frasi?"). Then the scale in figure 4.1 appeared detailing the meaning of values 1 ("very different") to 6 ("very similar"). This scale was used to avoid too many responses on a central, neutral value.

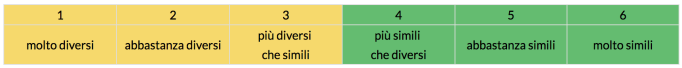


Figure 4.1: Scale used in human text similarity judgment task.

Models. We trained the models described in Chapter 1 on all 64 categories as well as on the 6 categories of the stimuli set.

Representational Similarity Analysis For the linear logistic regression models, we can obtain a feature vector for each sentence, the probability estimates. In this feature vector, each element is the probability the sentence belongs to each class (i.e., the class probability distribution).

In the deep neural networks we used, this corresponds to the final softmax layer. However, the softmax layers in these two deep neural networks are extremely sparse (which is not the case for the probablity estimates in the linear models) and the same RSMs can be computed from any layer. Therefore, we chose the final dense layers (of 512 features in LSTM and 128 in CNN) since they have a richer representation (i.e., more information) than the softmax layer.

The RSM for each model is the pairwise correlation between all feature vectors within a model. The human RSM is built by inserting the similarity scores from the experiment. Then these RSMs are correlated with each other to find the most similar to the human judgement RSM.

1.3 Results & Discussion

The results of models performance on the 6-way classification are in table 4.1. The same pattern of results are maintained as in the 64-way classification with the LSTM outperforming other models. It is possible that the results are biased to benefit LSTM clustering by using categories that were clustered in the LSTM model; however, this does not guarantee that the LSTM captures the specific similarities between these specific sentences more than the other models. To be safe, a more unbiased approach would be to use the categories that cluster for all models.

There is also a large jump in performance from top-1 to top-2 accuracy. The RSMs of each model are compared to human judgment. The RSMs for each model trained on the 64-way classification are in figure 4.2 for this stimuli set. The RSMs for each model trained on the 6-way classification are in figure 4.3. IN the latter, the similarities and disimalrities increase as expected since the classifiers have an easier task of differentiating the categories. The RSM of human judgments is in figure 4.4. The upper triangles of these RSMs are vectorized, normalized and correlated and the results

Model	Top-1	Top-2
Tf-idf	73.18	87.67
Avg. w2v	73.67	89.38
CNN	76.50	90.46
LSTM	**77.06**	**91.18**

Table 4.1: Top-1 and top-2 accuracy (%) on 6-way text

for the 64-way model are in figure 4.5a and the results for the 6-way model are in 4.5b (however, see figure 3.3 for RSM of RSMs on test set).

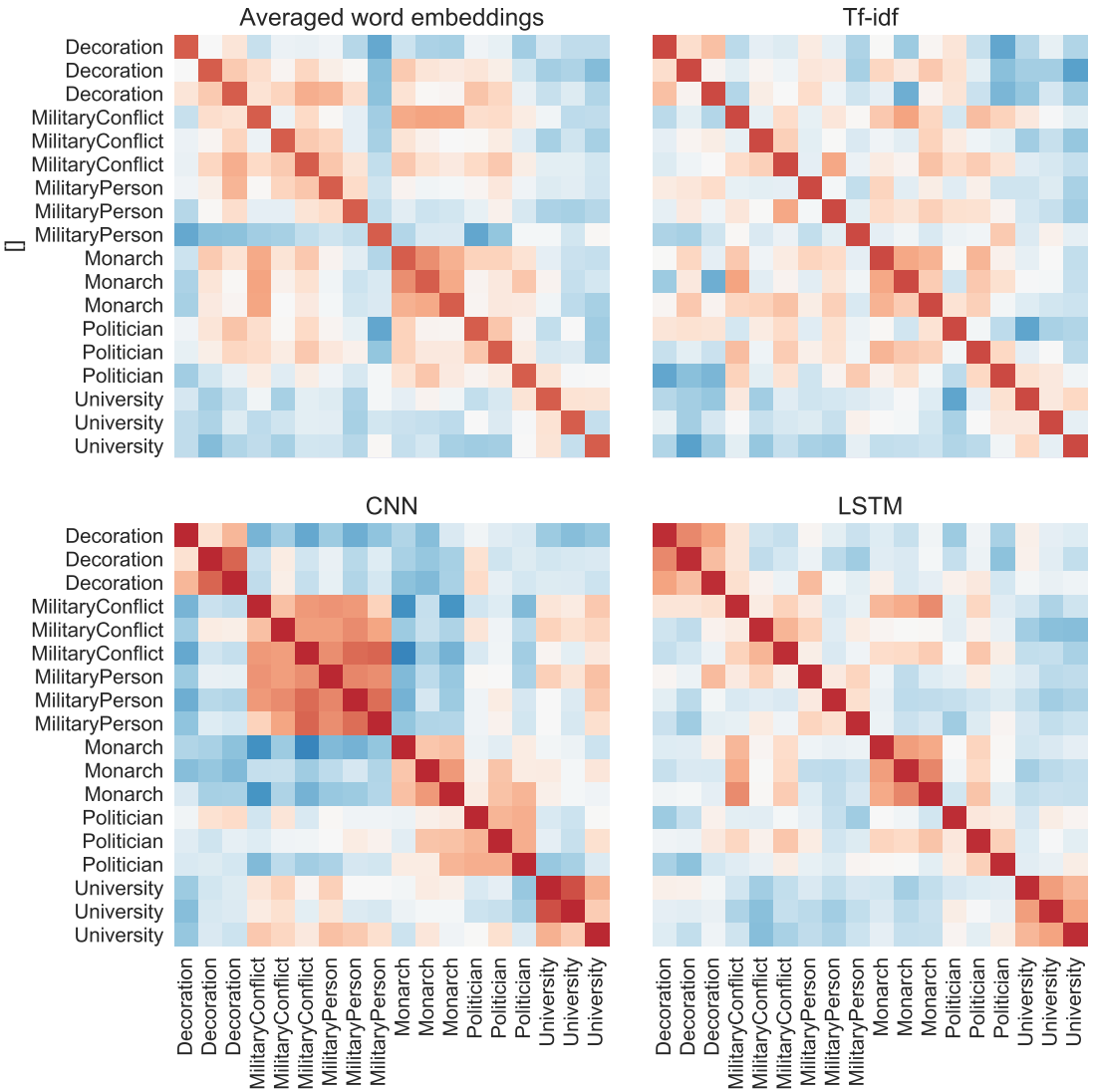


Figure 4.2: Representational similarity matrices for stimuli from models trained on 64-way classification. Each element is the correlation between two sentence feature vectors where 1.0= dark red and -1.0=dark blue. Each sentence is represented by its DBpedia category.

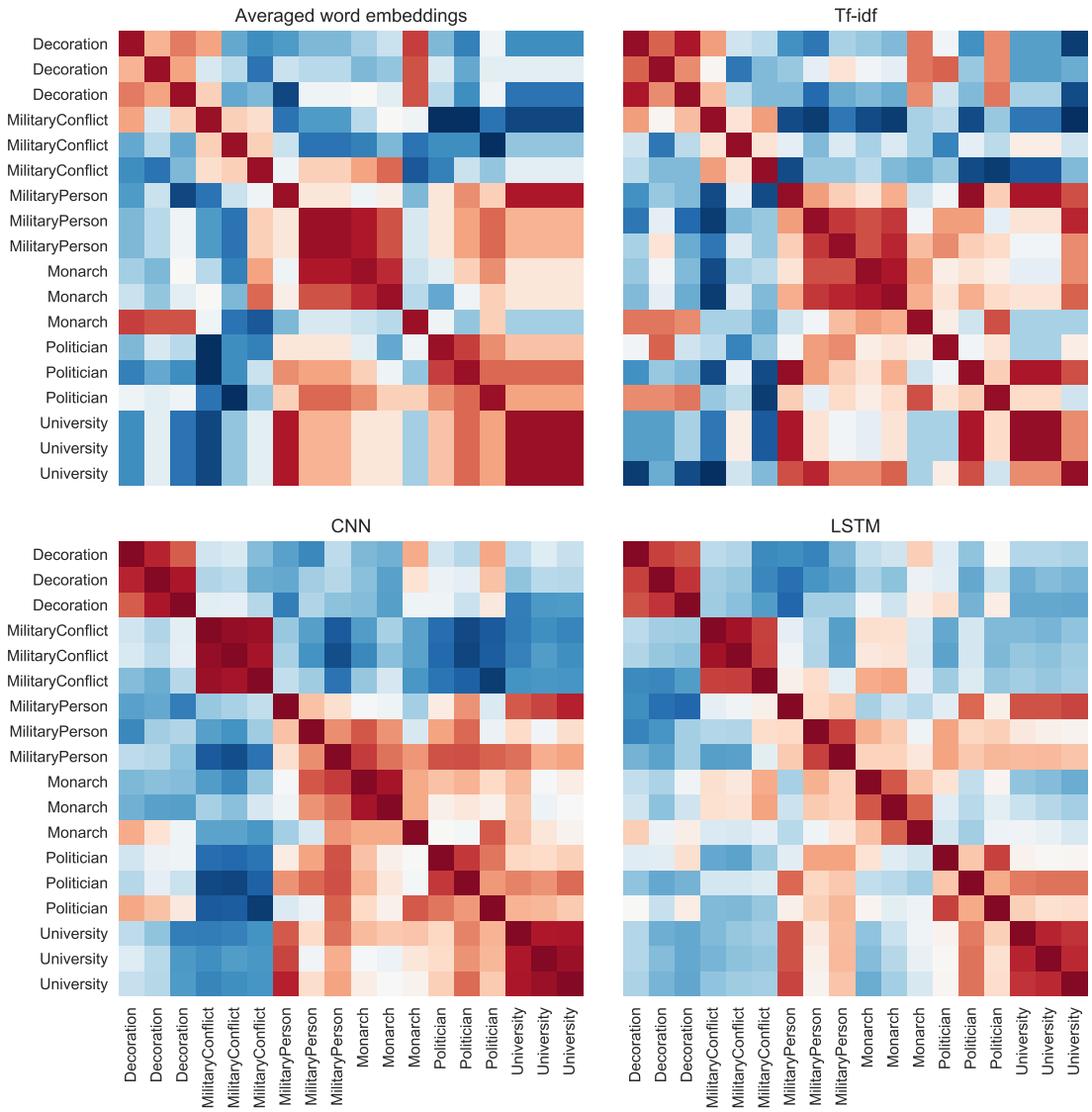


Figure 4.3: Representational similarity matrices for stimuli from models trained on 6-way classification. Each element is the correlation between two sentence feature vectors where 1.0= dark red and -1.0=dark blue. Each sentence is represented by its DBpedia category.

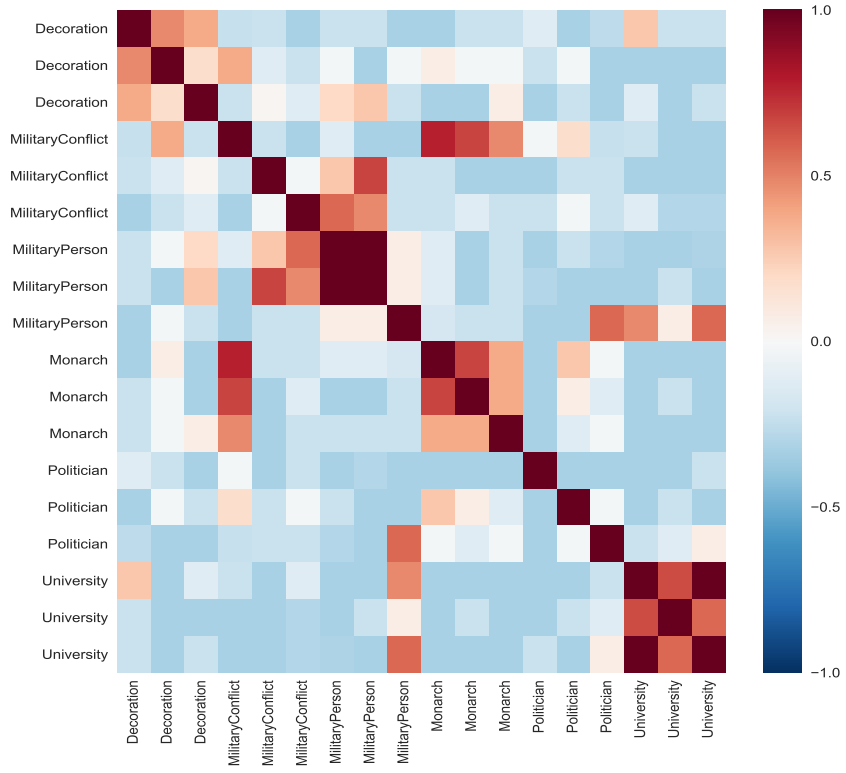
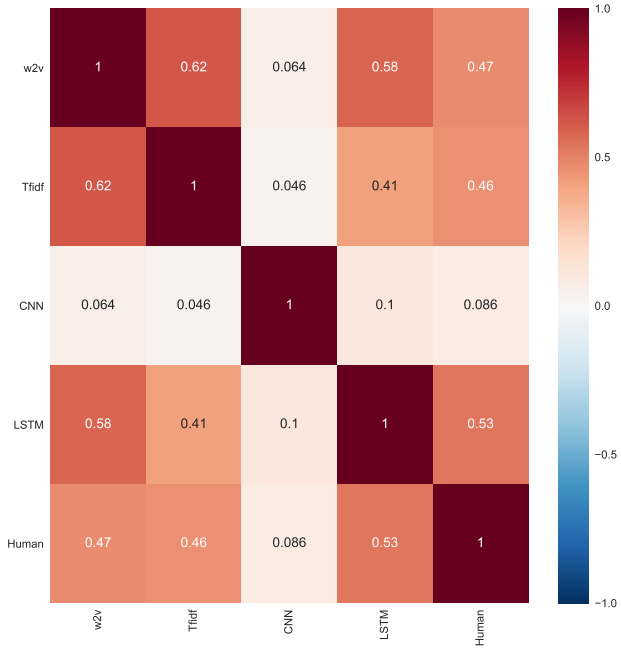
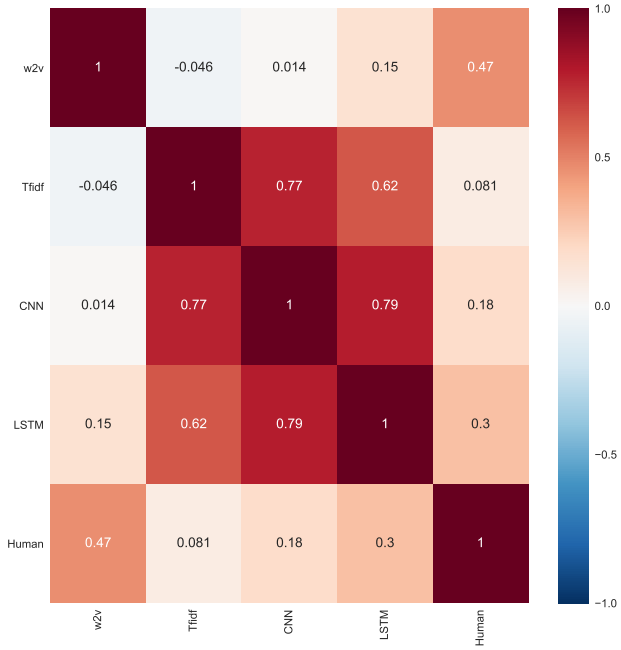


Figure 4.4: RSM of human judgment.



(a) 64-way classification model.



(b) 6-way classification model

Figure 4.5: RSM of model and human RSMs. Each cell is the result of taking the Spearman correlation between a RSM from figures 4.2 and 4.3 and the human RSM in figure 4.4. Only the vectorized upper triangles without the diagonals are correlated.

The human judgments were scaled from 1 and 6 to -1 and 1 to match other RSMs, which were normalized by subtracting the mean. The main result is that human similarity evaluations are best captured by the information within LSTM models in the 64-way classification feature vectors and within averaged word2vec models in the 6-way classification feature vectors. The first vectors are more meaningful for the purpose of the cognitive neuroscience experiment described in Chapter 3, since these are the ones that will be used for analysis (for further comparison between models see the section Representational Similarity Analysis in chapter 5). Here we learn that by comparing model similarity scores to human similarity judgment we can describe how much of this type of information is encoded in each model. Then, when each model is mapped to the brain (as described in Chapter 3), it is possible to see, for instance, that the CNN does not seem to capture human semantic similarity (although it may capture other linguistic or semantic features). Therefore, instead of assuming the CNN model is a model of semantics, we narrow its representation to exclude semantic similarity. However, these are preliminary results because this experiment should involve more categories to better capture judgment across the semantic space. It seems these results are biased by the specific categories chosen since the similarity between models is quite different when using a sample of the test set of all semantic categories as in figure 3.3: the similarity between models is not the same does not follow figure 4.5a. However, this experiment is a proof-of-principle for the interpretation approach we are presenting (see figure 1.1 for approach).

2 Experiment 3: Human Text Categorization

2.1 Introduction

Here we build a task that human's can do but machines cannot: we selected sentences that are classified differently by most models in Experiment 1 as a proxy for ambiguous sentences; that is to say, we chose sentences where the models are focusing on different information. The goal is to 1) have human choose the main category of sentences that are difficult to classify for models, creating a challenge for any model that attempts to capture human intuition on text categorization, and, 2) see which model is closer to human performance on difficult, ambiguous, or easily-confused sentences.

2.2 Methods

Subjects.16 native Italian participants were recruited through the Figure Eight platform. We requested at least 4 independent scores per stimuli (mean amount of judgments per worker = 29.5, SD = 24.5).

Stimuli. A dataset was created composed of sentences from four categories that were classified differently by at least three of the four models in Experiment 1. The four categories were: Album, Musical Artist, Song, and Musical Genre. For instance, the sentence "Testo e musica sono di Antonio Pagliuca ed Aldo Tagliapietra" [Text and music are by Antonio Pagliuca and Aldo Tagliapietra] was labeled as "Album" using the DBpedia label of the containing article, as "Song" by avg. word2vec embeddings and CNN models, "Music Genre" by the tf-idf model and "Album" by the LSTM (see Experiment 3 in Appendix for full dataset). These four categories were chosen because they are often confused in all models. Cohen's kappa for inter-rater agreement = -0.12, which is considered practically random and confirms that these models disagree on these sentences while they usually have high agreement.

Procedure. Sentences were presented randomly and participants had to select which of the four categories best described the sentence.

2.3 Results & Discussion

Since only four categories were used, we reran the models from Experiment 1 on a 4-way categorization task. See figure 4.2 for performance on 4-way classification task. The pattern changes from the 64-way classification: the CNN matches the LSTM in top-1 accuracy and surpasses it in top-2 accuracy, while tf-idf surpasses avg. word2vec embeddings.

Table 4.3 summarizes the main results. All models do substantially worse on this ambiguous dataset regarding both the DBpedia labels and human judgment. Even though LSTM performs best on this subset of sentences according to the DBpedia labels, using both 64- and 4-way classification feature vectors, tf-idf best captures human intuition under ambiguous circumstances, although all scores are quite low. In other words, when a sentence carries several possible categories, the information that tf-idf captures seems to be closest to what humans rely on (i.e., something similar to typical words of a category).

This task does not necessarily help with carachterizing different brain mappings (following the general goal in chapter 3), but helps us understand if the models capture the most difficult human intuitions on text categorization. Future work could adapt an

Model	Top-1	Top-2
Tf-idf	70.48	88.92
Avg. w2v	67.71	88.02
CNN	**71.88**	**90.35**
LSTM	**71.88**	90.33

Table 4.2: Top-1 and top-2 accuracy (%) on 4-way text categorization.

Model	DBpedia	Human judgment
Tf-idf	40.70	**45.35**
Avg. w2v	33.72	32.56
CNN	36.05	31.40
LSTM	**47.67**	34.88

(a) 64-way classification

Model	Dbpedia	Human judgment
Tf-idf	43.02	**45.35**
Avg. w2v	34.88	29.07
CNN	**45.35**	31.40
LSTM	39.53	34.88

(b) 4-way classification

Table 4.3: Accuracy on DBpedia labels and Human judgment using feature vectors obtained from model trained on 64 and 4 categories.

LSTM to use tfidf model when the softmax assigns a low maximum probability (i.e., when the model is uncertain). This could possibly capture human intuition best.

See tables for accuracy of the models on human judgment categorization. On this highly ambiguous stimuli set, DBpedia labels match human judgments 51.16% of the time. This low accuracy is expected for DBpedia lables on these sentences since they are quite ambiguous. As stated in the Dataset section of Chapter 2, DBpedia labels reflect the article's overall category, but not necessarily represent every sentence within the articles. If we want to capture whatever text categorization is for humans (i.e., what human's actually process), then models must capture the intuitive knowledge of a dataset like this.

5. Disassembling Machines

In this chapter we review a series of methods to analyze computational representations. This list is not exhaustive but allows us to tackle our main goal of showing that the four text categorization models from chapter 2 can be interpreted to show they are complementary. Complementary mean that they contain different information on semantic-linguistic features. Performance is not the only relevant metric on which to compare them. They hold different information and they can all be used to reveal new insights in brain decoding studies.

1 Experiment 4: Probing Animacy

1.1 Introduction

Probing consists of using feature vectors trained on one type of task (e.g., text categorization) to train a different task (e.g., animate-inanimate classification) in order to reveal if it contains the information of the latter task (Conneau et al., 2018). Conneau et al. (2018) offers a series of probing tasks, including sentence length, but since they are not highly semantic in nature and our data set is in Italian, we designed a new probing task. In this experiment, we labeled a subset of sentences from the test set in Experiment 1 as either "animate" or "inanimate". Then we train a linear classifier to see if these feature vectors obtained from categorization tasks have information to classify animacy. If this latter classification results in high accuracy, it means that the original feature vectors contain information on animacy.

The same can be done by labeling sentences with any other feature: degree of syntactic complexity, human/non-human, artificial/natural, transitive/intransitive, sentence length.

1.2 Methods

Dataset. A dataset was built by manually tagging the categories of the dataset in experiment 1 as either "animate" or "inanimate". Then the containing sentences of both categories were randomized and 20k were selected from each.
Since sentences within a certain category, for instance "dog", belong to articles about dogs, these are assumed to be animate. Closer inspection confirmed that they belong to the assumed category in approximately 90% of cases.
Using the models from Experiment 1, the sentences from the dataset were evaluated to obtain their feature vectors.

Models. Using the new animacy labels, these pre-trained feature vectors were used to traina linear support vector machine with scikit-learn default parameters[1]. The models previously used are marked ($\langle number \rangle$)

- **Tf-idf prob. (1):** Probability estimates (1x64 vectors) from Logistic Regression tf-idf features.
- **avg. w2v:** Averaged word2vec embeddings.
- **w2v prob.:**Product of Logistic Regression coefficient matrix (one row for each category) and the averaged w2v input.
- **w2v raw:**Product of Logistic Regression coefficient matrix (one row for each category) and the averaged w2v input.
- **w2v prob. (2):** Probability estimates from Logistic Regression averaged w2v features.
- **CNN conv1:** Conv layer 1 of the CNN, which represents local compositions of trigrams.
- **CNN conv2:** Conv layer 2 of the CNN, which represents intermediate compositions of conv-maxpool layer 1.
- **CNN conv dense final (3):** CNN's final dense layer of 128 features before softmax layer, representing the global composition of sentence's category.
- **LSTM dense final (4):** LSTM's final dense layer of 512 features before softmax layer, representing the cumulative composition of sentence's category.

1.3 Results & Discussion

See table 5.1 for results. The averaged word2vec embeddings trained on logistic regression had the most information on animacy. Each score can be a measure of animacy encoded in model's feature vectors. This means that averaged word embeddings and tf-idf contain more animacy representations than the CNN and LSTM. We also see that the CNN learns animacy gradually throughout the layers. This can be seen as the CNN needs to learn more animacy in order to classify. Therefore, if these models are decoded by brain data, one can describe how much animacy is being decoded. One could perform probing with other lexical, syntactic, or semantic features (e.g., dependency parsing, sentence length, place-tool distinction) to better describe the representations. This is important because once all four models are mapped to the brain, the information represented in each region can be better interpreted: the regions where only tf-idf and average word embeddings overlap can be said to contain more animacy than the regions where only CNN and LSTM overlap. Of course, it is not possible to seperate animacy from text categorization in a single model, therefore these regions encode both to different measurable degrees.

[1] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

Model	Anim.-inanim. clas.
Tf-idf prob. (1)	82.46
Avg. w2v	78.17
w2v raw	67.84
w2v prob. (2)	**84.05**
CNN conv1	71.01
CNN conv2	71.04
CNN dense final (3)	78.99
LSTM dense final (4)	78.80

Table 5.1: F1-score (%) on 2-way text categorization. F1-score was used because it was different than accuracy in this task. The models previously used are marked ($\langle number \rangle$)

2 Experiment 5: Distorting inputs and architecture

2.1 Introduction

By altering the inputs or changing the architecture, it is possible to evaluate how much of what is changed is contained within the original representation. For instance, if by scrambling word order, the model performs 5% worse than when having correct word order, then word order information is quite low than if it lost 25% accuracy. Since some models could use syntax or word order more than others, it is important to measure how much of this type of information is contained in each model.

Scrambling or distorting word order. Word order gives information about long- and short-distance dependencies, which is part of a syntactic knowledge. By randomizing word order, it is possible to see how much accuracy is lost, and therefore how much of the original accuracy was learning from normal word order.

Removing word embeddings. By removing word embeddings, the models have no previous "world knowledge" and must learn from initial random vectors. Therefore, the accuracy lost is how much of pre-trained word meaning is contained in the original model.

Removing layers. We will remove conv2 from the CNN to compare model complexity.

2.2 Methods

Dataset. Test set described in Experiment 1.

Models.

- LSTM with 1 layer: original LSTM from Experiment 1;
- LSTM with scrambled word order;
- LSTM without word2vec embeddings;
- multilayer CNN: original CNN from Experiment 1;
- multilayer CNN with scrambled word order;

- multilayer without word2vec embeddings;
- CNN with 1 layer while mainting all other parameters the same;

2.3 Results & Discussion

See table 5.2 for results. Whatever the models lose in accuracy from these distortions is thought to be the proportion of that feature contained in the original model.

Model	3 layers	4 layers	Scrambled	w/o word2vec
CNN	70.42 (+0.9)	69.52 (orig.)	67.74 (-1.8)	62.24 (-7.3)
LSTM	72.32 (orig.)	72.21 (-0.1)	70.75 (-1.6)	65.90 (-6.4)

Table 5.2: Accuracy (%) when distorting inputs and architecture. 3 layers = LSTM or conv-maxpool layer + dense + softmax. 4 layers = LSTM or conv-maxpool layer + second LSTM or conv-maxpool layer + dense + softmax. Between parenthesis, the difference between column model and original model is presented. Orig.: original model. Scrambled was performed on the original models.

We can see that both models are trained to not learn from word order since it was not optimal given the specific task of text categorization. Whereas a higher amount of their performance is due to using pre-trained word embeddings.

The stacked LSTM (4 layers) performs slightly worse than the single LSTM layer model (original model). The one-layer CNN performed slightly better (70.42% accuracy) than the multilayer model (69.52%), although neither modal is fully optimized. One of the issues is we wished to fix certain parameters a priori to capture certain type of compositionality. For instance, we set it to take small ngrams, trigrams in our case, instead of trying larger ngrams or multiple ngrams. However, so far, whatever the additional layer is learning in the multilayer, it is partially overfitting as it is not helping improve the generalization to the test set.

In conclusion, distorting inputs and architectures allows us to understand the undistorted models better. We learned how much information they contain on word embeddings and word order as well as how much accuracy is gained or lossed by altering the amount of layers. The latter informs the degree to which models may be overfitting.

3 Representational Similarity Analysis (RSA)

As seen in the previous chapter, RSA places multimodal representations (e.g., human, computer, brain data) in a common space. We were able to compare models to human judgments in Experiment 2. RSA also enables us to compare similarity between the models themselves to answer, for instance, how much of tf-idf information does a CNN carry? Figure 4.5a summarizes these results for the stimuli used for Experiment 2. For instance, the averaged word2vec model shares information with tf-idf at a correlation of rho=0.62, whereas the CNN does not correlate with tf-idf (rho=0.05). Therefore, it is possible to measure what is shared and not shared in the representations. Figure 3.3 takes a random subsample of 5% of the test set to compare models. Results change considerably. Results in figure 3.3 show that the two most similar models are the two deep neural networks while tf-idf and LSTM are the most disimilar. Therefore, it is

plausible to assume that Experiment 2 is not representative of the whole feature vector and the human judgment experiment should include more categories to make inferences on the full-category models.

3.1 Hierarchical Clustering of Representational Similarity Matrix

An important question is whether a model captures the human-intuitive semantic relationships between categories. In figure 5.1 we cluster the average similarity between categories. The results seem to be human-intuitive but should be confirmed by a human judgment task: humans would have to group the categories and then their grouping could be compared to this clustering by measuring the distance between categories.

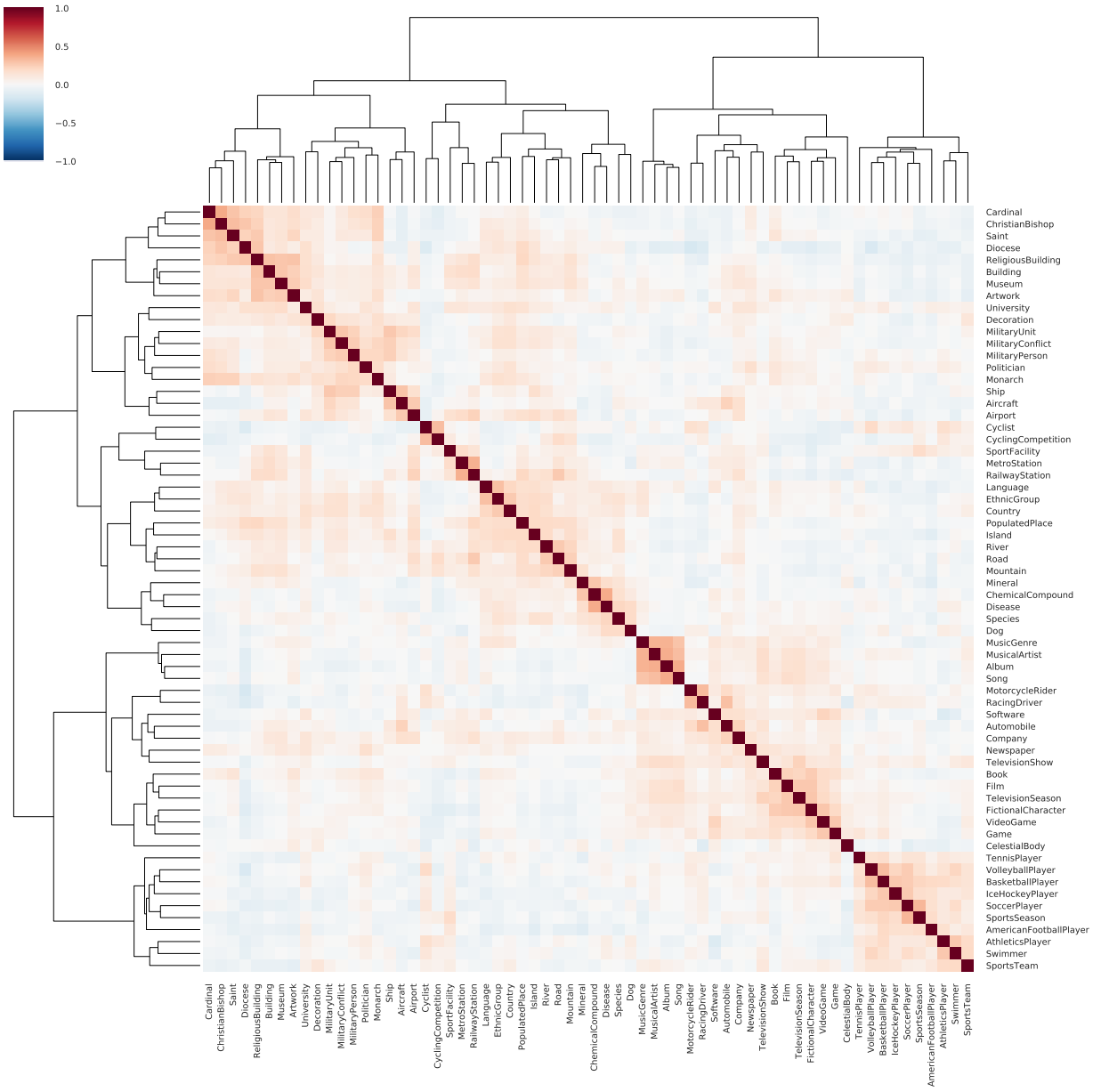


Figure 5.1: Each cell computed by taking the mean of the an RSM built of 20 random sentence feature vectors from two categories. For instance, on average, sentences from Cardinal and sentences from Christian Bishop tend to correlate at rho=0.47, and this value is placed in the corresping cell. Then all cells are clustered using Ward clustering and Euclidean distance.

4 Understanding Hidden Layers' Superficiality and Abstraction with Rouge

The method used to select the stimuli in Experiment 1 is an original way of finding representative samples of a given layer. Once obtained, it is possible to test the assumed superficial-to-abstract nature of CNNs by measuring how similar these sentences are among each other on a superficial level. To accomplish this, we use the Rouge metric (Lin, 2004), which counts the amount of overlapping units such as n-grams between two segments of text. This is generally used to compare machine and human translations and machine and human-generated summaries.

Table 5.3 shows that as predicted, conv1 sentences have a higher superficial similarity than dense final sentences. This means that the conv1 finds high similarity between sentences that are superficially the same whereas dense final is able to find high similarity between sentences with the same category even though they may be superficially more dissimilar than in conv1. For instance, using features vectors from either conv1 or the final dense layer, one cluster containing sentences originally labelled as "sports team" by DBpedia, resulted in two sentences that were highly similar superficially for conv1 feature vectors and more abstract for dense final feature vectors (sentences were translated):

Conv1 layer learns more superficial features:

- The Olympic Committee of Qatar was recognized by the CIO in 1980.
- The Olympic Committee of Monegasco was recognized by the CIO in 1953.

Final dense layer learns more abstract features:

- Valentyna Sevcenko carried the flag during the opening ceremony of the Olympics.
- The Shield then began to attack Kane, except to withdraw when the Usos intervened.

This is an interesting finding because it reproduces what is found in visual CNNs[2]. The CNN seems to be an abstraction machine. When the layers are mapped to the brain, we can say more about what each region encodes: layer 1 encodes more superficial representations and the final layers encode more abstract representations.

[2] e.g., http://cs231n.github.io/understanding-cnn/

	Conv1	Dense Final
Rouge 1	0.144	0.117
Rouge 1	0.060	0.020

Table 5.3: Rouge 1-gram and 2-gram scores between sentences within clusters from Experiment 1.

6. General Discussion

The goal of this thesis was to show the value of using complementary and interpretable computational representations of a given task for brain mapping. In Chapter 2, we described machines theoretically to show that they could provide different types and levels of text categorization since they compose meaning differently. In chapter 4 and 5, we used a series of techniques to interpret each models representations including comparing similarity measures to human judgments, probing, distorting inputs and architecture, RSA, and comparing Rouge metrics on representative sentences obtained through clustering.

In Table 6.1 we provide a summary of the relevance of different features for each model, a semantic-linguistic description of what each model encodes.

Feature	Tf-idf	Avg. w2v	CNN	LSTM
Exp. 1: DBpedia Text categorization	66	67	70	72
Exp. 2: Human Semantic similarity	46	47	9	53
Exp. 3: Human Ambiguous Text categorization	45	33	31	35
Exp. 4: Animacy	83	84	79	79
Exp. 5: Previous knowledge	98	98	11	9
Exp. 5: Syntax: word order	0	0	3	2
Fig. 4.5: Tf-idf	100	62	5	41

Table 6.1: Summary of relevance of different features for each model. Text categorization and semantic similarity scores are their accuracies (%); animacy is the f1-score (%) since it did not match accuracy in this task; or correlation (rho) value, depending on the feature. Previous knowledge and word order scores are calculated as follows: for the CNN and LSTM, the score = (score with condition - score without condition)/score with condition. It is is the proportion that removing the condition represents of the original condition (e.g., 11% of the accuracy is due to previous knowledge). For avg. w2v, one can assume that without this previous knowledge the model cannot learn beyond change (2%) and tf-idf also learns most of its knowledge before training the model, and therefore their scores are a theoretical 0.

Another important finding, described in section 4 of chapter 5, is that the CNN captures more superficial representations in the first layer and a more abstract representation in the final layers. This follows what has been shown for visual CNNs, but to our knowledge it is the first time it has been shown with text CNNs.

1 Building models of human text categorization vs. understanding brain regions

There are two different scientific questions in this thesis:

1. How can we build a computational model of human text categorization?
2. What information is in each brain region that can decode a computational representation?

It is possible to tackle each question separately or combine them. To answer question 1, we can compare models' predictions to human judgments as a gold standard. This was done in Experiments 2 using judgments on sentence similarity to see how much of this information is captured in the text categorization models. The LSTM model best captured human judgment. It was also carried out in Experiment 3, where we found that tf-idf best captures the human intuition of text categorization in difficult or ambiguous circumstances. As proposed in the Discussion of Experiment 3, these models could be combined to better match human intuition. On the other hand, the computational experiments of chapter 5 could be done with human judgment instead to asnwer this question. For example, the animacy probing task of Experiment 4 explains the animacy information contained in the representations (i.e., question 2) but a human judgment on the animacy of sentences should be done to see which model captures human's intuitions on animacy (i.e., question 1). A a perfect model of human intuition would achieve 100% accuracy on all human judgment tasks.

Question 2 can be answered by the computational experiments of chapter 5. We show it is possible to map models onto the brain (chapter 2) and by interpreting their semantic and linguistic properties (chapters 4 and 5), one can better understand and compare the resulting brain regions. In this sense, as long as certain models accurately capture information, they are useful and complementary for understanding brain functional activity. For instance, if probing is performed with other linguistic or semantic properties (e.g., dependency parsing, sentence length, place-tool distinction) the resulting brain regions can be better interpreted. In our case, after mapping the four models from chapter 1 on the brain, we would interpret that the regions where only tf-idf and average word embeddings overlap can be said to contain more animacy than the regions where only CNN and LSTM overlap. Therefore, even though a given model may not capture human intuition, it may still be decoded by brain activity, which can then be described.

Finally, these questions can be combined: one could build models to match human intuition using human judgment tasks as the ones in chapter 4 and then interpret the representations with methods from Chapter 5. Furthermore, many different semantic and linguistic features could also be measured to gradually match what psycholinguistics proposes humans use when comprehending or categorizing a sentence (David, 2000) by tuning the models. The hypothesis would be that a more psycholinguistically accurate model would better match human judgment.

2 Limitations

One limitation of this study is that models such as the CNN can be further optimized for performance, for instance by removing layers, but this would take away from

its multi-layer architecture which creates gradual compositionality. The benefits of the multilayer outweigh the cons. All complex models (e.g., deep neural networks) overfit to some degree since there are more parameters then theoretically needed to fit the task. The question of how each model overfits is not clear. What is clear is that feature vectors can decode brain data and a multilayer model is preferable for our goal of obtaining a model of gradual compositionality: it has more intermediate layers of representation which could allow us to map more levels of abstraction, from local compositionality and superficial representations to global and abstract representations. Having a partially wrong model is preferable than not having a model of that level of representation. Furthermore, we have found that the disimilarity between the first conv layer and the final dense layer is higher when there is an intermediate conv layer in the middle. From an fMRI analysis point of view, this would be an issue for signal-noise ratio in finding the separate contribution of the two layers to an external neural representation.

Another limitation is that the human judgment experiments should involve more categories to better capture judgment across the semantic space. It is plausible that the results are biased by the specific categories chosen since the similarity between models is quite different when using a sample of the test set as in figure 3.3. However, they are a proof-of-principle for the interpretation framework we are presenting.

3 Future work

An alternative task to text categorization is semantic textual similarity which trains models to measure the degree to which the underlying semantics of two segments of text are equivalent to each other or are paraphrases (Agirre et al., 2016). This would capture the other main form of semantic compositionalty which is the exact meaning of a sentence (instead of its category as done here).

Alternative models for either task that should reveal interesting insights are those incorporating 1) universal sentence encoders that intend to represent enough information to transfer well to many different natural language processing tasks (e.g., Conneau et al. (2017)), 2) syntax for semantic classification as is the Tree LSTM (Tai et al., 2015), and 3) better prediction of upcoming words such as a bidirectionalLSTM (Graves et al., 2013).

As mentioned before, alternative interpretation tasks include the SentEval package Conneau et al. (2018), but this can only be applied to English data sets.

Finally, the approach presented in this thesis (see figure 1.1 in Chapter 1) is viable not only for many different tasks that can be mapped onto the brain using fMRI, but also for different cognitive neuroscience methods. For instance, one could decode the different time step representations of an LSTM from MEG or EEG temporal. In this case, one could show how different brain areas are recruited as a whole sentence is gradually comprehended at the time-steps of words 1,5,15, for instance.

4 Conclusion

Whereas previous studies use a single model for decoding, encoding or searchlight analysis, we have shown the potential of using multiple models of the same task to map complementary forms of compositionality onto the brain. Furthermore, we have described a series of methods to interpret representations –from human judgment to

computational analysis. This framework allows us to understand the resulting brain regions function with greater detail.

6. Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abnar, S., Ahmed, R., Mijnheer, M., and Zuidema, W. (2017). Experiential, distributional and dependency-based word embeddings have complementary roles in decoding brain activity. *arXiv preprint arXiv:1711.09285*.

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., and Wiebe, J. (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Anderson, A. J., Binder, J. R., Fernandino, L., Humphries, C. J., Conant, L. L., Aguilar, M., Wang, X., Doko, D., and Raizada, R. D. (2016). Predicting neural activity patterns associated with sentences using a neurobiologically motivated model of semantic representation. *Cerebral Cortex*, 27(9):4379–4395.

Anderson, A. J., Lalor, E. C., Lin, F., Binder, J. R., Fernandino, L., Humphries, C. J., Conant, L. L., Raizada, R. D., Grimm, S., and Wang, X. (2018). Multiple regions of a cortical network commonly encode the meaning of words in multiple grammatical positions of read sentences. *Cerebral Cortex*.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Binder, J. R., Conant, L. L., Humphries, C. J., Fernandino, L., Simons, S. B., Aguilar, M., and Desai, R. H. (2016). Toward a brain-based componential semantic representation. *Cognitive neuropsychology*, 33(3-4):130–174.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6:27755.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. (2016). Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Crepaldi D, Amenta S, P. M. K. E. B. M. (2015). Subtitle-based word frequency estimates for italian. *Proceedings of the Annual Meeting of the Italian Association For Experimental Psychology Rovereto (Italy)*.

David, W. C. (2000). Psychology of language.

Etzel, J. A., Zacks, J. M., and Braver, T. S. (2013). Searchlight analysis: promise, pitfalls, and potential. *Neuroimage*, 78:261–269.

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (2014). Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1199–1209.

Gauthier, J. and Ivanova, A. (2018). Does the brain represent words? an evaluation of brain decoding studies of language understanding. *arXiv preprint arXiv:1806.00591*.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE.

Hagoort, P. (2013). Muc (memory, unification, control) and beyond. *Frontiers in psychology*, 4:416.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jain, S. and Huth, A. (2018). Incorporating context into language encoding models for fmri. *bioRxiv*, page 327601.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mineroff, Z. A., Blank, I., Mahowald, K., and Fedorenko, E. (2017). A robust dissociation among the language, multiple demand, and default mode networks: evidence from inter-region correlations in effect size. *bioRxiv*, page 140384.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195.

Montague, R. (1970). English as a formal language.

Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fmri. *Neuroimage*, 56(2):400–410.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pereira, F., Lou, B., Pritchett, B., Ritter, S., Gershman, S. J., Kanwisher, N., Botvinick, M., and Fedorenko, E. (2018). Toward a universal decoder of linguistic meaning from brain activation. *Nature communications*, 9(1):963.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Shalizi, C. (2013). Advanced data analysis from an elementary point of view.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363.

Zhang, W., Yoshida, T., and Tang, X. (2008). Tfidf, lsi and multi-word in information retrieval and text categorization. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 108–113. IEEE.

Zhang, Y. and Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

7. Appendix

1 Experiment 1 Sentence Examples

Category	Sentence
Monarch	Mori di leucemia nella capitale egiziana nell'autunno del 1988, all'eta di 67 anni.
Ship	A partire dal 1944 la nave fu utilizzata nel Pacifico per scortare le portaerei.
MilitaryConflict	I tedeschi, riconoscendo l'indifendibilita della propria posizione, si ritirarono dalla Marna verso nord.
Book	Fu rivisto e adattato agli altri racconti, quando fu inserito nella raccolta Mulliner Nights.
SportsSeason	La divisa casalinga era completamente rossa, a strisce verticali nere, pantaloncini neri e calzettoni rossoneri.
CyclingCompetition	Il tedesco Bert Grabsch domino la gara, mantenendo una velocita media di 50km/h.
EthnicGroup	Chiaramente, la tribu trae ancora grande orgoglio dalla sua prodezza e dal suo valore militari.
Museum	Il museo si trova in quella che un tempo era la centrale termoelettrica di Bankside.
TennisPlayer	Fino alla finale l'Errani, partendo dalle qualificazioni, ha incontrato solo tenniste americane battendole tutte.
Road	Il rimanente tratto, da Rezzato a Limone sul Garda, e invece rimasto all'ANAS.
Road	Superato lo svincolo di Cosenza, l'autostrada attraversa le varie montagne in direzione Altilia.
Dog	Il cane da orso della Carelia e un cane da caccia della famiglia degli spitz.
MotorcycleRider	E ottavo nel 1998 e ancora nono l'anno successivo, con una Yamaha YZF-R6.
Disease	Per quanto riguarda l'origine della viremia esiste la suddivisione fra forma attiva e passiva
MusicGenre	Questi nuovi complessi incorporavano gli elementi piu commerciali di alternative rock, emo e post-hardcore.

Category	Sentence
ReligiousBuilding	Il campanile e una torre pendente e infatti inclinato di circa 1,4 in direzione Est-Sud-Est.
Monarch	In mancanza di eredi, gli successe come era facile prevedere, il fratello minore Leopoldo.
Monarch	In mancanza di eredi, gli successe come era facile prevedere, il fratello minore Leopoldo.
Road	Il rimanente tratto, da Rezzato a Limone sul Garda, e invece rimasto all'ANAS.
RacingDriver	Anche nell'inverno 2008-2009 partecipa alla GP2 Asia, vincendola, e nel 2009 al campionato GP2.
TelevisionSeason	La serie e ambientata in Sicilia, ma e stata quasi interamente girata in Puglia.
TelevisionShow	Dal 2 maggio 2016 lo spazio dedicato al TG5 non va piu in onda.
Dog	Il Vastgotaspets e un cane di piccole dimensioni ma con un carattere molto forte.
Diocese	Il territorio della diocesi si estende su 43.110km2 ed e suddiviso in 4 parrocchie.
SportsTeam	Anche se era ovviamente per attaccare la Kru, Konnan ha rifiutato la fiducia di Kip.
Airport	L'autorita aeroportuale ha in concessione l'aeroporto dalla citta di New York dal 1947.
SportsTeam	Anche se era ovviamente per attaccare la Kru, Konnan ha rifiutato la fiducia di Kip.
SoccerPlayer	Poi Giuseppe Viani lo fa esordire in Serie A il 20 marzo 1960 in Udinese-Milan.
Newspaper	Negli anni novanta la rivista continuo ad essere un punto di riferimento del settore.
VolleyballPlayer	Per il campionato 2017-18 veste la maglia del Volley Milano, sempre in Serie A1.
Island	Diresse la piantagione con un pugno di ferro che porto al suo omicidio nel 1904.
Mineral	Il nome del minerale e stato attribuito in onore di Pavel'Ivanovich Stepanov, geologo russo.
Game	Il gioco e stato creato dal matematico Alain Rivollet e dall'informatico Gaetan Beaujannot.
VolleyballPlayer	Giulia Pincerato inizia la sua carriera pallavolistica nella squadra giovanile di Legnaro, nel 1999.
Disease	Per quanto riguarda l'origine della viremia esiste la suddivisione fra forma attiva e passiva
MilitaryConflict	Sull'altro versante, da Vienna si insisteva con Gyulai per una condotta piu energica.
TelevisionShow	L8 gennaio 2018 inizia la settima edizione sempre con la conduzione di Paolo Bonolis.
Ship	Mentre transitava nel canale Kaiser Wilhelm il 14 marzo 1917, la Kaiserin si areno.
Road	Superato lo svincolo di Cosenza, l'autostrada attraversa le varie montagne in direzione Altilia.

Category	Sentence
TelevisionShow	Questo format era gia stato sperimentato nel novembre 2007 ma abbandonato dopo poco tempo.
AmericanFootballPlayer	Nella settimana 8 fece registrare il primo intercetto stagionale nella vittoria sugli Atlanta Falcons.
VolleyballPlayer	Giulia Pincerato inizia la sua carriera pallavolistica nella squadra giovanile di Legnaro, nel 1999.
CyclingCompetition	Presero il via da Compiegne 186 ciclisti, 92 di essi portarono a termine la gara.
Diocese	Sede vescovile e la citta di Ica, dove si trova la cattedrale di San Gerolamo.
Game	Nel 1990 la TSR pubblico una seconda edizione del gioco, etichettandola erroneamente come terza edizione..
TennisPlayer	Agli US Open esce al primo turno di qualificazioni contro la greca Maria Sakkari.
Politician	Il mandato termino tuttavia pochi mesi dopo, a causa della vittoria elettorale della CDU.
Road	Il rimanente tratto, da Rezzato a Limone sul Garda, e invece rimasto all'ANAS.
ChemicalCompound	Il confronto con l'urea ricavata dall'urina lo convinse di aver sintetizzato quel composto.
CelestialBody	HD 30197 e una stella gigante arancione di magnitudine +5,99 situata nella costellazione del Toro.
ChristianBishop	Nel 1909 e stato insignito di un Master of Arts honoris causa dall Hobart College.
Software	MariaDB si compila sia su processori a 32 bit sia su processori a 64.
MusicGenre	La Frenchcore e un sottogenere musicale molto veloce e ritmato proposto soprattutto in Francia.
Politician	Il mandato termino tuttavia pochi mesi dopo, a causa della vittoria elettorale della CDU.
Book	Fu rivisto e adattato agli altri racconti, quando fu inserito nella raccolta Mulliner Nights.
SportsSeason	La divisa casalinga era completamente rossa, a strisce verticali nere, pantaloncini neri e calzettoni rossoneri.
TelevisionShow	L8 gennaio 2018 inizia la settima edizione sempre con la conduzione di Paolo Bonolis.
River	Scorre poi lungo il confine con Meduna di Livenza e la provincia di Treviso.
SportsSeason	Questa voce raccoglie le informazioni riguardanti la Pallavolo Chieri nelle competizioni ufficiali della stagione 2003-2004.
SoccerPlayer	Conclusa la carriera di calciatore, ha intrapreso quella di allenatore nelle categorie dilettantistiche lombarde.
Mountain	La vetta del Nelion fu scalata la prima volta da Eric Shipton nel 1929.
Film	By the Sad Sea Waves e un cortometraggio muto del 1917 diretto da Alfred J.
TelevisionShow	Dal 2 maggio 2016 lo spazio dedicato al TG5 non va piu in onda.

Table 7.1: Examples from dataset of Experiment 1.

1.1 Classification Report

	category	f1_score	precision	recall	support
0	Aircraft	0.77	0.82	0.73	1200.0
1	Airport	0.81	0.86	0.77	1200.0
2	Album	0.62	0.62	0.63	1200.0
3	AmericanFootballPlayer	0.92	0.96	0.89	1200.0
4	Artwork	0.69	0.64	0.74	1200.0
5	AthleticsPlayer	0.78	0.80	0.76	1200.0
6	Automobile	0.84	0.80	0.89	1200.0
7	BasketballPlayer	0.74	0.66	0.84	1200.0
8	Book	0.42	0.39	0.46	1200.0
9	Building	0.54	0.54	0.55	1200.0
10	Cardinal	0.60	0.54	0.68	1200.0
11	CelestialBody	0.95	0.94	0.95	1200.0
12	ChemicalCompound	0.82	0.88	0.77	1200.0
13	ChristianBishop	0.45	0.55	0.38	1200.0
14	Company	0.54	0.55	0.54	1200.0
15	Country	0.49	0.51	0.47	1200.0
16	CyclingCompetition	0.89	0.97	0.82	1200.0
17	Cyclist	0.84	0.81	0.87	1200.0
18	Decoration	0.88	0.89	0.87	1200.0
19	Diocese	0.86	0.87	0.85	1200.0
20	Disease	0.80	0.71	0.90	1200.0
21	Dog	0.89	0.92	0.86	1200.0
22	EthnicGroup	0.63	0.61	0.65	1200.0
23	FictionalCharacter	0.55	0.56	0.54	1200.0
24	Film	0.55	0.54	0.56	1200.0
25	Game	0.73	0.68	0.80	1200.0
26	IceHockeyPlayer	0.88	0.91	0.86	1200.0
27	Island	0.71	0.78	0.64	1200.0
28	Language	0.77	0.75	0.80	1200.0
29	MetroStation	0.84	0.80	0.88	1200.0
30	MilitaryConflict	0.43	0.54	0.35	1200.0
31	MilitaryPerson	0.48	0.54	0.43	1200.0
32	MilitaryUnit	0.56	0.51	0.63	1200.0
33	Mineral	0.92	0.92	0.91	1200.0
34	Monarch	0.56	0.50	0.64	1200.0
35	MotorcycleRider	0.87	0.88	0.87	1200.0
36	Mountain	0.76	0.83	0.69	1200.0
37	Museum	0.55	0.56	0.55	1200.0
38	MusicalArtist	0.59	0.57	0.61	1200.0
39	MusicGenre	0.78	0.75	0.82	1200.0
40	Newspaper	0.73	0.77	0.69	1200.0
41	Politician	0.54	0.50	0.59	1200.0
42	PopulatedPlace	0.56	0.63	0.51	1200.0
43	RacingDriver	0.83	0.84	0.82	1200.0
44	RailwayStation	0.85	0.86	0.83	1200.0

	category	f1_score	precision	recall	support
45	ReligiousBuilding	0.62	0.66	0.59	1200.0
46	River	0.81	0.79	0.83	1200.0
47	Road	0.86	0.84	0.89	1200.0
48	Saint	0.54	0.52	0.57	1200.0
49	Ship	0.76	0.71	0.82	1200.0
50	SoccerPlayer	0.75	0.73	0.77	1200.0
51	Software	0.88	0.86	0.90	1200.0
52	Song	0.62	0.67	0.57	1200.0
53	Species	0.82	0.81	0.83	1200.0
54	SportFacility	0.83	0.84	0.82	1200.0
55	SportsSeason	0.84	0.87	0.81	1200.0
56	SportsTeam	0.92	0.90	0.94	1200.0
57	Swimmer	0.80	0.86	0.75	1200.0
58	TelevisionSeason	0.54	0.56	0.53	1200.0
59	TelevisionShow	0.72	0.73	0.71	1200.0
60	TennisPlayer	0.92	0.93	0.91	1200.0
61	University	0.75	0.72	0.80	1200.0
62	VideoGame	0.55	0.71	0.46	1200.0
63	VolleyballPlayer	0.87	0.83	0.91	1200.0
	avg/total	**0.72**	**0.73**	**0.72**	**76800.0**

Table 7.2: LSTM classification report on 64-way classification.

2 Experiment 2

Stimuli for sentence similarity task.

Category	Sentence
University	L'Universita di Turku e la seconda maggiore universita della Finlandia per numero di studenti dopo l'Universita di Helsinki.
University	L'UCO ha partnership internazionali con piu di 75 universita del mondo.
University	Per entrambi i corsi master, il diploma finale e rilasciato da ALaRI - Universita della Svizzera Italiana, in collaborazione con l'ETH Zurich e il Politecnico di Milano.
Decoration	La croce d'argento puo essere concessa anche ad intere unita militari o navali, a civili e a citta.
Decoration	Sul retro del medaglione, invece, si trovava la figura di Massimiliano I, fondatore dell'ordine, circondato dalla scritta, sempre in dicitura antica.
Decoration	L'Ordine dispone delle seguenti classi di benemerenza
MilitaryConflict	La "Piacenza" pago lo scontro con 27 morti e 32 feriti gravi.
MilitaryConflict	Filippo V aveva avuto dal suo primo matrimonio tre figli ed era chiaro fine di Elisabetta Farnese ottenere ducati in Italia per i propri figli.
MilitaryConflict	I ministri Orlando e Sonnino, e con loro gli alti comandi alleati, sollecitarono ripetutamente Diaz perche desse inizio all'attacco risolutivo e il generale dovette piegarsi.
MilitaryPerson	Anderson si diplomo allo Staff College, Camberley nel 1928 ed entro nella 50th Division.
MilitaryPerson	Il 13 luglio intercettava con altri quattro Macchi una dozzina di Spitfire che stavano attaccando dei Messerschmitt 109 e abbatteva in successione due Spitfire, il secondo dei quali, pilotato dal Flight Sgt.
MilitaryPerson	Nel luglio del 1943 la 9a Armata di Model prese parte alla battaglia di Kursk la piu grande battaglia di carri armati di tutta la seconda guerra mondiale.
Politician	Nato e cresciuto nel New Jersey, dopo il college Norcross entro in politica con il Partito Democratico.
Politician	Scoperto per un suo errore ad'avere ancora contatti con i liberali siciliani, l 8 luglio 1850 fu bandito formalmente dal Regno delle Due Sicilie.
Politician	Come loro appartiene alla minoranza religiosa sciita dell'Alawismo.
Monarch	Quando Guglielmo ottenne infine il permesso di visitare la moglie, decise di portare il primogenito in Inghilterra, cosa non gradita da Guglielmina.
Monarch	Dopo la morte della sua prima moglie, Maria Emanuela d Aviz, Filippo, su consiglio del padre, decise di risposarsi con la trentasettenne Maria I d Inghilterra.
Monarch	Federico era il figlio maggiore del principe Filippo Giuseppe di Salm-Kyrburg e di sua moglie, Maria Teresa di Horn.

3 Experiment 3

Table 7.4: Stimuli for ambiguous text categorization task.

Sentences
Che divento un altra volta il singolo natalizio piu venduto in Inghilterra.
Alcuni rapper “interpretano” le parti di due personaggi diversi, che dialogano l’un l’altro nella stessa canzone.
Nel febbraio 1997, gli U2 pubblicarono il singolo “Discotheque”, come brano di lancio del nuovo album.
I Cure arrivarono anche ad inserire un estratto di tale dibattito nella traccia finale dell’album.
Anche i Blondie realizzarono nello stesso periodo un brano simile, “Rapture”.
In Italia “Scars” e conosciuta grazie ai The Fire che l’hanno rifatta in una versione piu rock.
Quando Bjork ritorna, si unisce alla comitiva per una speciale performance della versione “Audition mix” della canzone.
Brian Wilson, lo storico membro dei Beach Boys, partecipa ai cori della canzone che da il nome al disco.
Maple, Xiu Xiu, Man Man, The Fiery Furnaces, e TV on the Radio.
Egli ha migrato particolarmente verso la Svezia poiche affascinato dalla ”buona” musica degli Europe, Roxette e ABBA.
Il 23 aprile viene pubblicata una versione acustica anche per il singolo ”Starring Role”.
Benche indiretta, anche ai Rolling Stones e ascritta un influenza sullo ”sleaze”.
Altre volte fu suonata durante la militanza di Page con i Black Crowes nel 1999.
Sul disco French si occupo anche di suonare la chitarra in qualche traccia.
Ha composto canzoni con Nick Manasseh, Future Cut e Feng Shui per l’album di debutto ”Turned on Underground”.
E il secondo brano con cui Mogol vince il festival ligure dopo ”Al di la”.
”Bump” viene anche utilizzato nella colonna sonora di ”Entourage”.
Il suo singolo successivo ”Don Quixote” fu l’ultimo singolo della striscia positiva ad entrare nella top 20 inglese.
Rapidamente divenne una delle canzoni piu popolari del XX secolo.
L’autrice invece si accompagna in questo disco con strumenti come chitarra acustica, banjo, ukulele e tastiere.
Il CD ebbe un ottimo successo grazie alla sua carica innovativa dovuta all’uso seppur ancora molto blando delle tastiere.
Hobsbawm, storico e docente inglese, con il libro ”The Jazz Scene” del 1961, e Amiri Baraka con ”Blues people”.
Nel 1977 ha rappresentato la Finlandia all’Eurovision con la canzone ”Lapponia”.
Howard in molte canzoni dei Manilla Road, cosi come dei Crimson Glory.
Guest star che parteciparono all’incisione di quest’album furono Richard Page dei Mr. Mister e Kevin Cronin dei REO Speedwagon.
Breakbot inizialmente riscosse un buon successo di pubblico grazie ai suoi remix.
In ”Tentacled” ritroviamo Silvia Chicco in un brano del 1989 originariamente inciso con Monica Cioce.
Alcuni dei suoi singoli classici sono reperibili anche oggi, ma ”Choice of Version” e una delle migliori raccolte.
”Cantabrasil” ripercorre praticamente tutta la storia moderna della musica brasiliana.

Pochi giorni piu tardi usci un 45 giri contenente le prime due canzoni "Mi sono innamorato di te" e "Angela".
Dylan volle dare un suono austero al tutto in sintonia con il contenuto dei testi.
Tipici del gruppo uno stile frenetico, estremamente ritmato e la breve durata delle canzoni.
"Lady Liberty", cambia registro con la comparsa dei fiati che rendono l'atmosfera molto piu energica.
Quest ultimo realizza per lui la strumentale per il singolo "Raise Up".
Fra i ballerini si puo riconoscere anche Cris Judd, ex-marito della cantante.
Bill lo volle come frontman e chitarrista dei Blue Grass Boys.
Dr. Dre con la canzone "What's the Difference", presente nell'album "The Chronic" del 2001.
Qualcuno nei media, in toni celebrativi, dichiaro la disco "morta" e il rock rianimato.
Fortunatamente riusci a riprendersi in tempo per le sessioni di registrazione di "Keep the Faith".
A meta febbraio 2006, "Wasteland" e arrivato in testa alle classifiche di Alternative e Modern Rock.
Nell'aprile del 1988 gli Scorpions pubblicarono il loro decimo album in studio, "Savage Amusement".
Il primo singolo estratto, "The Fly", spiazza i fan degli U2 degli anni ottanta.
"For What It's Worth" vendette circa un milione di copie nei soli Stati Uniti, diventando presto disco d'oro.
Fairies Wear Boots e l'ultimo brano presente nell'album Paranoid del gruppo heavy metal britannico Black Sabbath.
Con la quale introduce e spesso chiude, la maggior parte dei brani musicali a cui partecipa.
La quarta traccia e secondo singolo promozionale, "Walking on Air", e stata prodotta da Klas Ahlund.
La canzone vinse anche nella categoria "Singolo dell'Anno", ai Juno Award del 2003.
Harris, fan di musica "country", sposta lo stile di Joan verso il country-rock piu complesso di "David's Album".
"Love Me" debutto nel Canadian Hot 100, dove ha trascorso dodici settimane non consecutive nella classifica canadese.
Si tratta della pubblicazione di maggior successo del quartetto, tanto da essere piu volte ripubblicata da Taang!
Il brano e uno dei due della band a contenere poliritmie, il che e molto raro nella musica popolare.
Nel 2000, Carey ha ripubblicato la canzone con nuovi live vocali.
"Mr. Jones", una dei Talking Heads sul loro album del 1988 "Naked", similarmente descrive un Mr. Jones in terza persona.
Ha scritto o coscritto infine diversi brani per Tina Turner.
Compare insieme a Dylan nell'album del 2012 "Tempest" e nel 2015 nell'album "shadows in the night".
The Distillers e l'album d esordio del gruppo Punk Rock The Distillers.
In quest ultimo lavoro, e soprattutto molto apprezzato l'uso che Plunkett fa delle tastiere.
In alcuni frangenti il coro degli altri membri del gruppo si unisce alla voce di Robert Plant.
Sempre nel 1997 esce la ristampa di "Testa plastica", contenente la cover della canzone dei Violent Femmes "Gone Daddy Gone".
Testo e musica sono di Antonio Pagliuca ed Aldo Tagliapietra.
Diversamente da quanto avveniva normalmente, si decise di fare un disco di canzoni, quindi non solo recitato.
"Jumpin Jumpin" e entrato nelle classifiche di molti altri paesi, tra cui Svezia, Belgio e Francia.

La compilation del 2004 "Join the Dots B-Sides & Rarities 19782001" utilizza anch essa il titolo "Lovesong".
Il libro traccia i profili di alcuni musicisti outsider conosciuti e ha ispirato due CD compilation, vendute separatamente.
Alla chitarra egli suona con esuberanza, spesso percuotendo con "slap" le corde.
I Mascarimiri sono un gruppo musicale italiano composto da 4 elementi, provenienti dalla citta di Muro Leccese.
Con "Solo noi" e "Innamorati" Toto inizia a scrivere anche i testi delle proprie canzoni, oltre alla musica.
Nel 1977, Giorgio Moroder e Pete Bellotte hanno prodotto "I Feel Love" per Donna Summer.
Un gruppo chiamato E-rotic ricevette l'attenzione con testi sessualmente provocanti e video musicali.
Le peculiarita di questo brano sono il ritmo semplice, veloce e ipnotico.
Sierra suona la chitarra, il flauto ed e la voce principale del duo.
Un altro momento ironico del disco cantato in inglese.
Changes e una nota canzone del gruppo heavy metal britannico Black Sabbath.
La versione country di Ricky Nelson fu un successo da Top 40 hit negli Stati Uniti.
Il titolo della canzone ha anche ispirato il nome del movimento artistico degli anni ottanta "Memphis Group".
Dopo essersi separato dagli Orb, il membro Jimmy Cauty pubblico "Space", mentre Paterson compose il singolo "Little Fluffy Clouds".
Nel 2000 fu messo in commercio lalbum "Quadros modernos" e il testo "Livrao da Musica Brasileira".
L'idea dell'album nasce da un commento rilasciato da John Lennon che, quando pubblico il suo singolo "Instant Karma!"
E diventato famoso grazie ad artisti come Kid Cudi, Donald Glover, Kid Sister, Azealia Banks e Chiddy Bang.
La loro prima pubblicazione ufficiale, l'EP "Metal & Dust", e datata febbraio 2013.
Il singolo vendera circa 70.000 copie ma non gli varra nessuna certificazione.
Nel marzo 2004 Tobias registra la canzone "Decks-Athron" con DJ Krush e Tatuki Oshima, inserita nell'album "Jaku".
Molte copertine degli album dei Minutemen come "Paranoid Time", "What Makes a Man Start Fires?"
Harrison fece ristrutturare la chitarra e la uso per la copertina dell'album.
La prima fonte on-line aveva annunciato che il titolo dell'album sarebbe stato "Bogey Depot".
La musica ha un ritmo caraibico, accompagnato nell'arrangiamento da una slide guitar.