SmartTitle: An Intelligent Subtitling System for Second Language Learning

Abstract

SmartTitle is a system designed to increase one's vocabulary in a foreign language. A user watches foreign media with subtitles shown in their native language with select words obscured in the text by replacing them with asterisks (***). A neural network is employed to rank and select words that are deemed good candidates to obscure based on a set of proposed features. The intent is that the system promotes active listening to dialogue and strengthens phrase associations by guiding the user to deduce the meaning of the obscured words.

1 Introduction

Learning a second language can be an immensely rewarding process. Most people in the world speak at least two languages and as globalization increases, knowing another language can only benefit one's career. Canada, in particular, is a country that recognizes two official languages. An understanding of both French and English can be advantageous (even in primarily English speaking regions) and greatly increase one's appreciation of the cultural heritage across the country.

While language learning can be fun, it is often the case that after a certain level of basic introduction, learners can feel as though their progress has slowed and that continued learning requires significant effort. Aside from the subtleties of grammar and language structure, arguably the most time consuming and important aspect of reaching a comfortable level of control in using a new language is building vocabulary. Reading text in a foreign language can seem tedious because it requires frequent pauses to lookup new words in a dictionary or consult other translation services.

Film and television are both wonderfully rich with language learning content. Spoken dialogue between characters exposes learners to common expressions and phrases while also providing proper pronunciation. In addition, the fact that television and film are presented as video, visual cues from scenes and can add strong contextual hints about the nature of the spoken dialogue.

SmartTitle is a language learning tool for vocabulary building that aims to augment the rich language content of television and film with a modified subtitle file in the learners native language. As English is the most widely spoken language in Canada, SmartTitle currently focuses on improving French vocabulary for English speakers. The system analyses an English subtitle file for a French television show or film and obscures words that are deemed good candidates to learn. A motivating example is taken from the show *How I Met Your Mother* and shown below:

Ted: So my plan is to take her to the roof. Barney: What is so great about the roof? Lily: Oh the roof is magical, you can see the moon and the stars, it's very romantic.	⇒	Ted: So my plan is to take her to the ****. Barney: What is so great about the ****? Lily: Oh the **** is magical, you can see the moon and the stars, it's very romantic.

Figure 1: SmartTitle Example Modification

In this example, we can observe that the characters are having a conversation about the roof. Each character mentions the word during this conversation and in fact, later in the episode, the characters move to the roof to discuss further. The word *roof* is therefore an excellent candidate to obscure.

SmartTitle employs a neural network to compute a scoring function z, that takes an input of 6 computed word features, $f_{1:6}$, which consist of integers and real numbers as discussed in section 3 and outputs a candidacy score as a real number within the range $[1, 5]$. The function is designed such that a score of 1 corresponds to a poor candidate to obscure and a score of 5 to an excellent candidate.

$$z : (f_{1:6}) \rightarrow \mathbb{R} \in [1, 5] \tag{1}$$

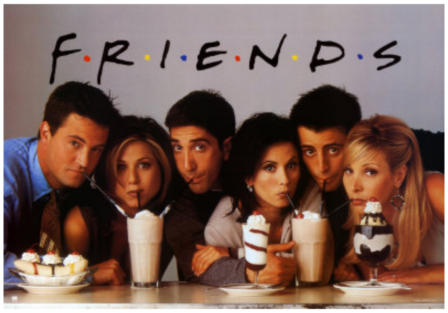
With this scoring function, SmartTitle is able to rank words within a subtitle and select the best word to obscure. In our example, it would be expected that the word *roof* would result in a high score and that it be obscured. In this way, the hope would also be that the viewer, while listening to the French dialogue, would come to understand the French word for roof : *toit*.

This report discusses the details related to training a feed-forward neural network using supervised learning for computation of the scoring function. Generation of the training data will be discussed as well as the selection of the input word features. Several alternative neural network constructions are evaluated and finally a discussion about the results of the system is presented.

2 Data Collection

For supervised learning to be possible, a dataset must be created that provides an expected score for each pattern of word features in the set. A learning algorithm can then be applied to find a solution that minimizes the error of the predicted scores [1].

For SmartTitle, this dataset was generated manually. The training and test datasets combined consisted of approximately 2000 manually scored words, taken from three episodes of the first season of the sitcom *Friends* dubbed in French with corresponding English subtitles.



```
11
00:00:29,260 → 00:00:32,332
We like the comedian...

12
00:00:32,540 → 00:00:34,735
It's just that that's not...

13
00:00:34,940 → 00:00:36,851
...why we bought the ticket.
```

Figure 2: *Friends* Subtitle Dataset.

Fortunately, the SubRip (.srt) file format is very simple to work with and a python library (aeidon) had already been written to parse and modify subtitle data. Each file simply contains, in plain-text, a list of subtitles where the first line is the subtitle index, the second line is the time that the subtitle should be shown, and the subsequent lines are the subtitle text as show above in figure 2

Words are tokenized from the subtitle text using whitespace as a delimiter. The range of score values [1,5] was selected to facilitate the dataset generation process. The words in the subtitles were all ranked by one person with moderate understanding of French. The following simple, however subjective, rules for scoring were followed:

1 : very poor, uninteresting word, extremely difficult to understand from context
2 : poor word, difficult to understand, or uncertain about word
3 : understandable from context, word not necessarily interesting
4 : easy to understand from context, fairly interesting, not best in subtitle
5 : easy to understand from context, very interesting, best in subtitle

A decision was also made to perform an initial filter for words that are not very likely to be good candidates for learning. This in turn also simplified the ranking process. SmartTitle uses the natural language toolkit (nltk) in python to filter the words for only a select subset of the parts of speech tags. The parts of speech that SmartTitle is concerned with are nouns, adjectives and verbs. Other parts of speech known as function words (of, it, the, in) are considered poor candidates and ignored by the system. The tagger in nltk takes as input a list of strings and outputs the most likely part of speech tag based on an n-gram model constructed from a large corpus of English text. The use of n-grams is discussed further in section 3.3. In addition, tokens containing numerals, punctuation, or any capital characters are also filtered. Words containing capital characters are filtered because they very often correspond to proper nouns, meaning names of places and people (eg Paris, Chandler, New York), which should never be obscured.

Table 1: Good POS Tags Subset

TAG	DESCRIPTION	EXAMPLE
JJ	Adjective	*big*
JJR	Ajective, comparative	*bigger*
JJS	Adjective, superlative	*biggest*
NN	Noun, singular	*dog*
NNS	Noun, plural	*dogs*
VB	Verb, base form	*eat*
VBD	Verb, past tense	*ate*
VBN	Verb, past part	*ate*
VBP	Verb, non-3rd pers. sing, present	*eat*
VBZ	Verb, 3rd pers. sing, present	*eat*

As an example consider the list of word tokens below. A part of speech tag is found for each word in the list and then filtered so that only words with tags in table 1 remain:

```
input = [’We’, ’know’, ’how’, ’cruel’, ’parents’ , ’can’ ’be...’ ]
filtered =  [ (’know’,’VBP’), (’cruel’,’JJ’), (’parents’,’NNS’) ]
```

3 Word Features

The technique of generating features on words is well studied in text analysis literature [2], [3]. For the purposes of SmartTitle, a feature can be thought of as any property of a word that may be relevant for determining its score. These features can be real-valued, integer-valued or range over an arbitrary, fixed set of symbols [2]. SmartTitle employs 6 features, $f_{1:6}$, and a discussion of each follows in the subsequent sections.

3.1 Part of Speech Index

As discussed in the previous section, the part of speech tags are very likely important features to consider in the scoring of the words. Therefore SmartTitle simply takes the index of the words tag in the array of all valid tags as shown in table 1.

$$f_1 = \text{validTags.indexOf}(tag)$$

3.2 Adjacent Turn Frequency

Conversation between characters provide viewers with a lot of contextual information about the conversation subject. Work in text summarization identifies contributions by each person involved in a conversation as *turns* [4]. From the motivating example in the introduction, it was shown that in each turn, the characters added information about the conversation subject (in this case the *roof*)

SmartTitle adopts a simple measurement for a words conversational relevance by counting its frequency in the current, previous and next subtitles.

$$f_2 = \sum_{i=cur-1}^{cur+1} \text{SUB}_i.\text{freq}(word)$$

3.3 Word Context Probability

N-grams are models on the sequence of words in natural languages. They are probabilistic models that can be used for predicting subsequent words or computing the likelihood of subsequent words, and take the form of an $(n-1)$ order Markov model [3]. For SmartTitle, it would be desirable that words that are obscured are more likely given the context of the subtitle so that viewers may more easily determine the obscured word. SmartTitle uses a 3-gram, generated from the Brown corpus of English text to compute f_3 as shown:

$$f_3 = P(word_i \mid word_{i-1}, word_{i-2})$$

This simply uses the direct probability of the current word, given the previous 2 words in the same subtitle as returned from the constructed 3-gram .

3.4 Relative Word Position

Where the word occurs within the subtitle was also deemed a relevant feature for determining its score. This feature was included allow the system to prefer to obscure for example words that occur near the end of a subtitle. If the word considered occurs at the i^{th} position in the current subtitle, f_4 is computed as shown:

$$f_4 = \frac{i}{\text{len}(\text{SUB}_{cur})}$$

3.5 Total Frequency

Here the total frequency of the word in the entire subtitle file for the show or film is considered. This provides SmartTitle, in a basic sense, how important or common a word is in the episode or film.

$$f_5 = \sum_{i=1}^{N} \text{SUB}_i.\text{freq}(word)$$

3.6 French Translation Edit Ratio

The English language shares a large number of words with French. These words are spelled and nearly pronounced the same, so for an English speaker, these words are considered less interesting to learn. For example, it is not very interesting to know the French word for *destination* is *destination*.

A common distance metric in natural language processing is the Levenshtein edit distance. The LD between two strings is defined as the minimum number of edits needed to transform one string into another, where an edit is defined as a character insertion, deletion, or substitution [7].

$$f_6 = \frac{\text{LD}(word, translation)}{\text{len}(word)}$$

To provide SmartTitle with some knowledge as to how similar a word is to its French translation, f_6 is computed as the Levenshtein distance of the word to its French translation, divided by the length of the word. The translation for each word is found by using the *Microsoft Bing* translation APIs.

4 Neural Network Construction

Feed forward neural networks are an established technique for solving regression problems for non-linear functions. Input features are combined linearly and fed into non-linear activation functions, for example the sigmoid function or the tanh function [5]. These values form the hidden values of the network, and there may be 1 or more of these hidden layers. In addition, bias terms can also be introduced to the network at each level. In the case of SmartTitle, the only parameters that were decided upon initially were the number of input features (6) and the number of outputs (1). The diagram below presents the basic feed forward neural network structure that was sought:

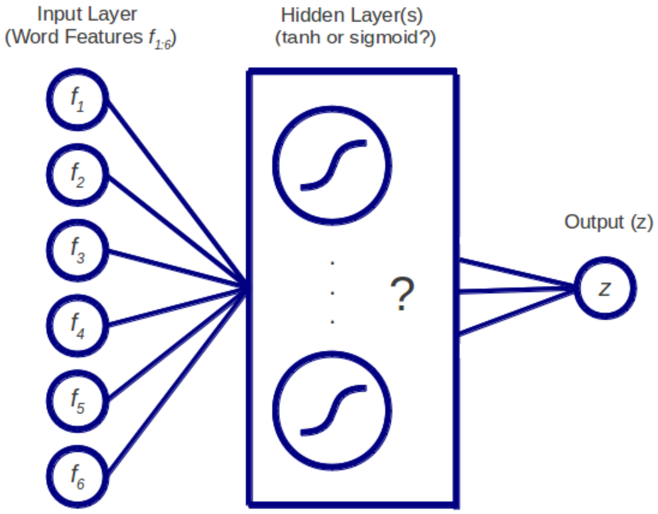


Figure 3: SmartTitle Neural Network Structure

With all these options for defining a network, it seemed daunting to decide which parameters and functions should be used for its definition. In the case of SmartTitle, an evaluation of several possible networks was conducted. The options varied in each network were the number of hidden units (2-5), the number of hidden layers (1-2), and the activation function used (tanh or sigmoid). A bias term was also added to the input and hidden layer(s).

Fortunately, PyBrain, a python library for building, training and evaluating neural networks, has recently been released [6]. With PyBrain, it was quite simple to construct, train and evaluate all variants of the neural network. Training required 75% of the collected dataset as explained in section 2. The networks were trained using the back propagation algorithm and the residual sum of squares as an error function. The remaining 25% of the dataset was reserved for testing each network. Each network was trained by iterating over 50 epochs of the data.

5 Results

After training the variations of the network, each was evaluated on the test dataset. A condensed summary of the networks and their modified parameters along with the mean squared error resulting on the train and test sets is shown in table 2. The best results from representative classes have been included (several of the weaker 2-layer results have been left out).

The results indicate that on average, the best network will score words about 1.2 points off of their true value. This may seem like a significant amount of error ($\sim$ 25%), however the goals of SmartTitle are more qualitative, relating to whether or not interesting words are obscured and learning actually takes place.

To experiment with the system, the neural net with the best min-max error (bolded in table 2) was selected for evaluation. For qualitative analysis, a selection of the remaining episodes in the first season of *Friends* were watched using SmartTitle modified subtitles. The highest scoring word was only obscured in a subtitle if it scored over 3.5.

After watching several episodes, the viewer was able to learn several “interesting” new words, some of which included: bouleversant (upsetting), plaquer (to dump), pourboire (tip) and cramé (cremated, see figure 4 for the scene that lead to this understanding).

There are several issues with the system that limit its usefulness. For one, the English subtitles do not always correspond to what is being said in French exactly. That is, the translation of the dialogue may change the meaning of what is said slightly or completely if it is better said another way in French. Secondly the system as it is does not always choose good words to obscure and further

Table 2: Comparison of Neural Networks Results

H. Layers	H. Units(1)	H. Units(2)	Actv. Func.	Avg. Train Error^2	Avg. Test Error^2
1	2	N/A	sigm	1.567	1.468
1	3	N/A	sigm	1.657	1.586
1	4	N/A	sigm	1.609	1.522
1	5	N/A	sigm	1.655	1.725
1	3	N/A	tanh	1.641	1.492
1	4	N/A	tanh	2.329	2.300
1	5	N/A	tanh	1.721	1.600
2	3	2	sigm	1.563	1.562
2	**4**	**1**	**sigm**	**1.532**	**1.511**
2	5	1	sigm	1.547	1.487
2	3	1	tanh	1.694	1.714
2	4	1	tanh	1.657	1.634
2	5	1	tanh	1.834	1.788

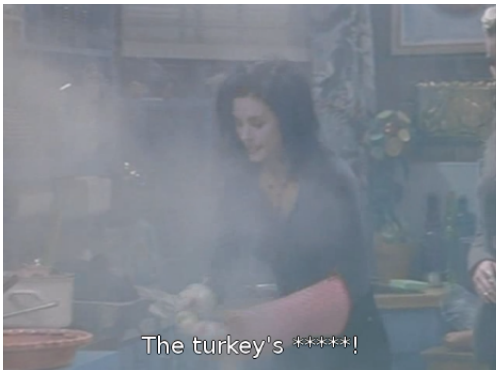


Figure 4: SmartTitle System

work could be done to investigate using a corpus of subtitle files instead of the English fiction, which definitely would have a different style and vocabulary when compared to spoken dialogue. Further work could be also be done exploring different word features or other regression techniques. As it is, and after only very brief evaluation however, the system does seem to create a novel learning experience and indeed promote paying more attention to the spoken dialogue.

References

[1] Duda, R., Hart P., Stork, D. (2001) *Pattern Classification (2nd edition)*, New York: Wiley.

[2] Turney, P. (1999) Learning Algorithms for Keyphrase Extraction. *Journal of Information Retrieval* 2(7):303-336, National Research Council Canada.

[3] Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Conference on Empirical Methods in Natural Language Processing* : 216-223. Sapporo.

[4] Carenini, G., Murray G., Ng R. (2011). *Methods for Mining and Summarizing Text Conversations* , Morgan and Claypool.

[5] Bishop, C. (2006). *Pattern Recognition and Machine Learning* , New York : Springer

[6] Schaul, T. et al (2010). PyBrain. *Journal of Machine Learning Research*

[7] Wikipedia contributors. Levenshtein distance (2011). *Wikipedia, The Free Encyclopedia.* Wikimedia Foundation, Inc.