# Leveraging plot summaries for video understanding

Ugo Jardonnet
jardonnet@lrde.epita.fr

September 22, 2010

**Abstract**

One of the main challenges for video indexing and retrieval is how to precisely annotate videos in the temporal and spatial domain: who does what, when, and where. Several researchers have considered using screenplays and closed captions to aid in related tasks such as naming people and retrieving actions (Everingham et˜al., 2006; Laptev et˜al., 2008; Cour et˜al., 2008, 2009). By combining appropriately screenplays and closed captions, we know who says what, and when. However, in many cases, such information is not available (Sankar et˜al., 2006, 2009). This project report addresses the problem of video annotation using short movie summary.
We investigate a comprehensive framework including information extraction from plot summary, on-demand classifier training based on plot summary, text to text and text to video alignment for automatic movie annotation using plot summary.

Supervisors:
Timothee Cour, Ivan Laptev, Josef Sivic


Ecole Normale Superieur de Cachan
Master Mathematics, Vision, Learning (MVA)


Willow - Computer Vision and Machine Learning Research Laboratory

# Contents

The film opens to eight men eating breakfast at a diner. Six of them wear matching suits and are using aliases: Mr. Blonde (Michael Madsen), Mr. Blue (Eddie Bunker), Mr. Brown (Quentin Tarantino), Mr. Orange (Tim Roth), Mr. Pink (Steve Buscemi), and Mr. White (Harvey Keitel). Among them is Los Angeles gangster Joe Cabot (Lawrence Tierney), and his son, "Nice Guy" Eddie Cabot (Chris Penn). Mr. Brown discusses his comparative analysis on Madonna's "Like a Virgin", Joe's senior moments involving his address book rankle Mr. White, and Mr. Pink defends his anti-tipping policy until Joe forces him to leave a tip for the waitresses. ...

Figure 1: Instance of Synopsis: Reservoir Dogs

# 1 Introduction

In this project, we will investigate the use of plot summaries for video indexing in movies and TV series. Plot summaries convey condensed information about the content of a video, ranging from detailed scene descriptions (example here) to coarser summaries with just a few sentences (example here). The main differences with a screenplay are A) the lack of dialog elements (we don't know who says what), and B) the lack of time stamps which in the case of screenplay can be automatically infered from closed captions. However, we do have information about the sequences of actions, scenes and events (who does what how and where) which could be useful for alignment. The goal is to align actions and events depicted in the plot summary to time intervals in the video. Plot summaries are much more widespread than screenplays. For example, the Imdb website alone references 1.3 million movies/TV episodes, 0.3 million plot outline/summaries as well as other useful side information such as images of actors. Wikipedia is another great source of plot summaries.

A number of cues can be used to align the plot summary and the video, such as temporal ordering (allowing for dynamic time warping algorithms), scene categorization, person recognition, dialog snippets that can be aligned to closed captions, recognizable actions, and much more. Our project consists of several independent tasks: text processing to extract formatted actions/events from plot summaries and retrieve automatically data from Imdb, vision techniques for scene categorization, action categorization, time of day classification.

In the following sections, we will make a clear distinction between the plot summary, or **synopsis**, which is a short summary of the movie (1 or 2 pages maximum, see Figure˜1), and the **screenplay** which provides a much more comprehensive description of movie content in terms of scenes, dialogs, events as well as camera motions (usually dozens of pages, see Figure˜2).

Based on successful previous works on screenplay to movie alignment (Everingham et˜al., 2006; Laptev et˜al., 2008; Cour et˜al., 2008, 2009), we could use the result of automatic alignment between screenplay and movie in order to

```
<description ascore="0.000" begin_frame="400"
end_frame="400" end_time="00:00:00" labels=""> Eight men
dressed in BLACK SUITS, sit around a table at a breakfast
cafe. They are MR. WHITE, MR. PINK, MR. BLUE, MR. BLONDE,
MR. ORANGE, MR. BROWN, NICE GUY EDDIE CABOT, and the big
boss, JOE CABOT. Most are finished eating and are enjoying
coffee and conversation. Joe flips through a small address
book. Mr. Pink is telling a long and involved story about
 Madonna. </description>
<speaker>MR. BROWN </speaker>
<monologue ascore="0.947" begin_frame="500"
begin_time="00:00:04" end_frame="700" end_time="00:00:12"
labels=""> "Like a Virgin" is all about a girl who digs a
guy with a big dick. The whole song is a metaphor for big
 dicks. </monologue> ...
```

Figure 2: Instance of Screenplay: Reservoir Dogs

evaluate results of synopsis to movie alignment. The validation process follows the following steps:

1. A0, Synopsis to screenplay alignment.
2. A1, alignment between screenplay and movie (our ground truth).
3. A2, Synopsis to movie alignment.
4. In order to evaluate the alignment between the synopsis and the movie (A2), we compare the alignment between the synopsis and the movie (A1), with the alignment between the synopsis and the screenplay (A0).

Note that the result of step A1 is obtained from previous work (Laptev et˜al., 2008).

This project report focus on the specific problem of scene alignment between plot summary and movie. After describing the different source of information and data used for the project section˜2, we introduce the theory of sequences alignment in section˜3. Then, section˜4 discusses the synopsis to screenplay alignment procedure. Finally, section˜5 depicts the entire framework of synopsis to video alignment.

# 2 Data

Natural language processing as well as video understanding requires some "world" knowledges. Most of the time important training sets are needed. Information retrieval requires huge amount of information about words and their related meanings and uses. This section describes the various datasets used for this project.

## 2.1 Synopsis and Screenplay

In order to obtain a set of synopses and screenplays, we developed two python scripts able to automatically retrieve such data from publicly available databases. Synopses have been fetched from Wikipedia and Imdb allowing us to constitute a dataset of 250 synopses of the best-movies Imdb list. Any Imdb synopsis can actually be downloaded across the 0.3 million plot summaries available on the website.
Our scripts make use of Imdbpy, a python API for the Imdb database, and Nodebox a python library dedicated to web page processing and parsing.
Screenplays, less easily available, were mainly obtained from
`http://www.moviescriptsandscreenplays.com/` and from the Willow laboratory database.

## 2.2 Semantic Information

An ontology is a model of some knowledge represented by a set of concepts and a set of relationships between those concepts. WordNet and Wiktionary are freely available ontologies.

**WordNet** WordNet 3.0 has been developed at Princeton University. It is a large lexical database of English, developed by linguists and computer scientists (Fellbaum, 1998). Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. The ontology provides relationships of synonymy, hypernymy (the relation of class to subclass, like animal to cat), hyponymy (subclass to class), holonymy (whole to part), meronymy (part of the whole), antomymy and much more. WordNet 3.0 contains about 150,000 words organized in approximately 117,000 synsets for a total of 206,941 word-sense pairs.
The database can be used to evaluate the semantic similarity between words (how much the sense of a word is different from another). Wordnet can also be used for word classification using hyponymy/hypernymy relationships (cat is an animal if animal is an hypernym of cat).
We implemented a C++ API allowing in-memory access to WordNet. Compared to on-demand WordNet parsing (as it is usually provided), the library offers a significant runtime speedup (up to 400 times) (Jardonnet, 2010). We

used this library for *semantic similarity computation* and *word categorization*.

**Wiktionary** The online ontology Wiktionary is an open-source dictionary storing lexical and semantic relationship between words. The database refers about 175,000 words and provides common semantic relationship like synonymy, hypernymy, or antonymy, but also knowledge related information like etymology, translation, quotation. Previous work already showed the quality of Wiktionary as a lexical semantic resource (Zesch et˜al., 2008b; Krizhanovsky and Lin, 2009). Wiktionary can be used, as Wordnet, for semantic similarity evaluation and word categorization. The website itself can be difficult to parse manually though but some libraries provide API to access the database (Zesch et˜al., 2008a). We didn't use Wiktionary for this project but we think it could worth using Wiktionary instead of WordNet in our framework, as Wiktionary is a constantly evolving platform.

## 2.3 Visual Information

The vision part of this project focused on the problem of scene classification. That is the ability to determine if a specific sequence of frame in the movie is happening in a street, a forest, indoor etc. In supervised learning, large learning set are required to achieve good recognition rates. We make use of 3 different datasets. We used ImageNet and Holliwood 2 in order to evaluate our classifier and the SUN dataset as a basis for automatic on-demand classifier trainer (see section˜5). However, each of these datasets seems reliable enough to be used conjointly as learning sets. Indeed, a classifier could benefit from the different quality and image type across the datasets.

**ImageNet** ImageNet (Deng et˜al., 2009) is a sibling of WordNet. The objective of imageNet is to provides visual illustration for most of the WordNet synsets (currently only the nouns). The database provides about 11,230,000 images organized in 15589 synsets. Images in this dataset are of different format, type and quality, close to the kind of results you can get on Google image, excluding false positive.

**Holliwood 2** Hollywood-2 dataset contains 12 classes of human actions and 10 classes of scenes distributed over 3669 video clips and approximately 20.1 hours of video in total. The dataset intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings. The dataset is composed of video clips extracted from 69 movies, it contains approximately 150 samples per action class and 130 samples per scene class in training and test subsets. A part of this dataset was originally used in the paper "Actions in Context" (Marszalek et˜al., 2009). Hollywood-2 is an extension of the earlier Hollywood dataset.

**SUN** The SUN database (Xiao et˜al., 2010), where SUN stands for Scene UNderstanding, is a large dataset containing 130,519 images in 899 categories. The different categories include indoor (Living, Bathroom, Church, Temple ...), Urban environment (Street, Road, Buildings ...) and Nature (Lake, Sky, Meadow ...). Images from this dataset are of "good" quality (maybe too good compared to common movie frameS), without obstruction and important image resolution.

# 3 Sequence Alignment

The theory of sequence alignment has been mainly developed for genetics where sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Typically, the problem consists of transforming one sequence into another using edit operations that replace, insert, or remove an element.
A large variety of algorithms exist to address the sequence alignment problem, in a major part of them dynamic programming is essential. Other approach use heuristic or statistical methods.

We associate a cost to each edit operation. The goal is to find the sequence of edits with the lowest total cost. The problem can be stated as a recursion, a sequence A is optimally edited into a sequence B by either:

1. Inserting the first character of B, and performing an optimal alignment of A and the tail of B
2. Deleting the first character of A, and performing the optimal alignment of the tail of A and B
3. Replacing the first character of A with the first character of B, and performing optimal alignments of the tails of A and B.

The partial alignments can be tabulated in a matrix, where cell $(i, j)$ contains the cost of the optimal alignment of $A[1..i]$ to $B[1..j]$. The cost in cell $(i, j)$ can be calculated by adding the cost of the relevant operations to the cost of its neighboring cells, and selecting the optimum.

We used a dynamic time warping approach (Ney, 1992) which allows only edit operation up to a certain size (e.g. maximum $k$ deletions). With $k$ small enough (in our case 5), the algorithm becomes tractable for large sequences.

# 4 Synopsis to Screenplay Alignment

## 4.1 Introduction

The goal here is to align two sequences of words, the synopsis and the screenplay. Two questions must be considered at this point:

- What is a word?
- Shall we use every words?

**What is a word?** The process of chopping up a sequence of caractere into pieces is called tokenization. The basic approach if you want token to be "words" is to chop on whitespace and throw away punctuation characters. Sadly, even for English there are a number of tricky cases.

- Apostrophe for possession and contractions: Which tokenization for isn't? isn't, is | n't, isn | t ...
- Collocations: San Francisco, Los Angeles should be considered as a single token. New York .
- Hyphenation: co-education, Hewlett-Packard must be one token but advertisements for air fares may contains something like "San Francisco-Los Angeles".

Detection of collocation can be performed efficiently using prefix trees (Ramabhadran et˜al., 2004).

**Shall we use every words?** Some words are not relevant for text alignment like *a, the, about, do ...* as they may be present anywhere in the text. This words are called stop words. Such words must be considered as noise since aligning a "the" at the beginning of the synopsis with a "the" at the end of the screenplay has only poor chance to be relevant. Moreover, the length of the two sequences is significantly different. It can be important to restrict alignment to "important" terms: places, peoples, actions ... in order to minimize misalignment.

In order to align a text 1 with a text 2 there is two possible solutions:

1. Alignment following exact matches between words from text 1 and words from text 2.
2. Alignment following best matches, based on a similarity function, between words from text 1 and words from text 2.

Two words must have a great similarity if they refer to the same thing, the problem of semantic distance between two words is discussed just after this introduction in subsubsection˜4.2.3.

## 4.2 Text Feature

This section discuss some of the text features used for this project. subsubsection˜4.2.1 introduce low frequency words. These words can be used exclusivly for text alignment in order to limit the length of the two sequences to meaningful/outstanding terms. subsubsection˜4.2.2 describes the concept of named entity. subsubsection˜4.2.3 explains how to define a semantic similarity measure between two word senses.

#### 4.2.1 Low Frequency Words

A very simple approach to identify outstanding terms is to rank them according to their relative frequency in English. terms of low frequency are uncommon and usually contains important information. They may be usefull for the alignement process. First step is to normalize the form of the word (e.g. *runners* becomes *runner*). This step is called lemmatization. Then stop words (i.e. words with very high frequency in english (a, the, but ...)) are removed. Then, the top 40% of lowest frequency words is kept in the sequence.

1. Lemmatization
2. Stop words reduction
3. Frequency estimation based on English corpus.

#### 4.2.2 Named Entity

Named entity recognition is the process of classifying words, or group of words, into categories like *person, organizations, locations, time of the day,* etc. State-of-the-art NER systems like Illinois NER (Ratinov and Roth, 2009) or the Stanford Named Entity Recognizer (Finkel et˜al., 2005) usually learn classes from large amount of manually annotated data (CoNLL datasets, Penn Treebank, ).

For video annotation, this information can be used to identify reference to movie characters, places (and these way scene like beach, street or river to extract from the video) or weather and night/day information easy to tag in the movie. Current NER system can be quite performing. However ambiguity issue can arise (like noun-entity ambiguity): The plural word jobs and the surname Jobs is an example of this problem (Nadeau et˜al., 2006).
Moreover such systems cannot be efficient without prior knowledge on the subject of the text processed.

A good method to efficiently retrieves places or celebrities name for instance is simply to possess a huge database with "every" possible places or celebrities. This technique cannot be perfect though. For instance most of the states in the USA have a river called by the very same name of the state. Sometimes it is possible to resolve the ambiguity based on the context. Sometimes this is very difficult. In "La statue de César au carrefour de la Croix-Rouge" César is obviously the french sculptor as opposed to the Roman Caesar (César in french) but you need world knowledge to make this decision. Sometimes it is just impossible to decide.

The detection of named entity can also use a prefix tree, as in section˜4.1, for efficient lookup in a knowledge base (Ramabhadran et˜al., 2004).

In practice the benefit of the disambiguation procedure is quite low (Chen et˜al., 1999) for common NER system. Some techniques different than a simple

lookup table (Nadeau et˜al., 2006) can be used to improve the precision but a good recall is important in our case since we want to detect the more we can. Indeed if something (a scene category or an actor) is also detected in the movie, we can later use this information as an anchor in the alignment process. However if the information is not in the movie, we can simply discard the named entity so that too much named entity should not be a problem.

#### 4.2.3 Semantic Distance

We define the semantic distance between two synsets on the hypernymy graph of the English WordNet, where each synsets is a possible sense of a word and the hypernymy relationship is the relation between the more general and the specific (e.g. animal is an hypernym of cat).
The distance between a synset $s_1$ and one of its hypernym $s_1^+$ (e.g. distance between *cat* and *animal*) is the shortest path between $s_1$ and $s_1^+$.

$$d(s_1, s_1^+) = \text{shortest\_path}(s_1, s_1^+)$$

Note that a synset can have two or more hypernyms. The shortest path problem can be solved with a Dijkstra's algorithm or a simple breadth first search (Dijkstra, 1959).

Let $h$ be the lowest common hypernym of two random synsets $s_1$ and $s_2$. The distance between $s_1$ and $s_2$ is the distance between $s_1$ and $h$ plus the distance $s_2$ and $h$.

$$d(s_1, s_2) = d(s_1, h) + d(s_2, h)$$

In Figure˜3, *carnivor* is the lowest common ancestor (hypernym) of *cat* and *dog*. $d(\text{cat}, \text{carnivor}) = 2$, $d(\text{dog}, \text{carnivor}) = 2$, so $d(\text{cat}, \text{dog}) = 2 + 2 = 4$.

Figure˜4 and Figure˜5 expose the code of semantic similarity computation. The code is written in C++ and the hierarchy is implemented using the Boost Graph Library (Siek et˜al., 2001). The function Hypernym_map returns the distance between a synset s and all its hypernym. This function uses the dijkstra algorithm and performs efficiently due to the limited number of levels in the hypernym hierarchy (distance maximum 10 hypernyms between a given synset and the top of the hierarchy). The function Semantic_distance in Figure˜5 find the lowest common ancestor in the hierarchy and return the semantic distance between *synset1* and *synset2*. The double loop in this function seems surprising, but you have to remember that the total number of hypernyms for both synsets is very limited and approximately constant.

# 5 Synopsis to Video Alignment

## 5.1 Goal

The goal here is to detect in the text if the action is happening in a particular scenery. We called possible scene classes like beach, forest, street etc concepts

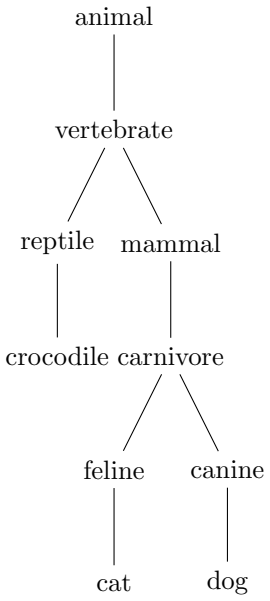



Figure 3: Example a hypernym hierarchy

(later a concept may cover persons, objects etc.). First concepts are extracted from the synopsis. Then, scene classifiers are trained for those concepts. Finally, the classifiers are applied to every frames of the movie. By comparing confidence values returned by these classifiers, we are able to assign a scene class (possibly none) for each frames of the movie. This gives us a sequence of scene class that we can align with the list of concepts previously extracted from the synopsis.

## 5.2 On Demand Classification

### 5.2.1 Concept Extraction

In order to detect evocation in the synopsis of possible scene category in the movie, we propose a python script based on the WordNet ontology. This script take a synopsis as input and output a list of concepts. We call concepts a class for which we can train a video classifier.

For instance the concept **Forest** exists in "I live near a forest", where the concept is introduced here by the word *forest*. But it also exists in "I'm going through the woods", through the word *woods*.
Now, shall we detect **Forrest** in "I roam between the trees"? Some cases are complex, and require complex context analysis. A "top" in English can be a tent but the word *top*, of course, does not always refers to a tent.

```
std::map<vertex, int>
hypernym_map(vertex s)
{
  std::map<vertex, int> map;
  boost::graph_traits<G>::out_edge_iterator e, e_end;
  std::queue<vertex> q;

  q.push(s);
  map[s] = 0;
  while (!q.empty())
  {
    vertex u = q.front(); q.pop();
    int new_d = map[u] + 1;
    for (tie(e, e_end) = out_edges(u, fg); e != e_end; ++e)
    {
      vertex v = target(*e,fg);
      q.push(v);
      if (map.find(v) != map.end())
      {
        if (new_d < map[v])
        map[v] = new_d;
        else
        q.pop();
      }
      else
      map[v] = new_d;
    }
  }
  return map;
}
```

Figure 4: Compute distances between a synset s and all its hypernyms.

```
int semantic_distance(const synset& synset1,
                      const synset& synset2)
{
  vertex v1 = synset1.id;
  vertex v2 = synset2.id;

  std::map<vertex, int> map1 = hypernym_map(v1);
  std::map<vertex, int> map2 = hypernym_map(v2);

  // For each ancestor synset common to both subject synsets,
  // find the connecting path length.
  // Return the shortest of these.

  int path_distance = -1;
  std::map<vertex, int>::iterator it, it2;
  for (it = map1.begin(); it != map1.end(); it++)
  for (it2 = map2.begin(); it2 != map2.end(); it2++)
  if (fg[it->first] == fg[it2->first])
  {
    int new_distance = it->second + it2->second;
    if (path_distance < 0  new_distance < path_distance)
    path_distance = new_distance;
  }

  return path_distance;
}
```

Figure 5: Compute the semantic similarity.

See example of concept extraction in a the synopsis of the movie Big Fish in subsection˜A.1.

### 5.2.2 Visual Feature

**Spatial HOG** First, histogram of oriented edges (HOG) descriptors are densely extracted on a regular grid at steps of 8 pixels. HOG features are computed using the code available online provided by Felzenszwalb et˜al. (2008), which gives a 31- dimension descriptor for each node of the grid. Then, $2 \times 2$ neighboring HOG descriptors are stacked together to form a descriptor with 124 dimensions. The stacked descriptors spatially overlap. This $2 \times 2$ neighbor stacking is important because the higher feature dimensionality provides more descriptive power. The descriptors are quantized into 300 visual words by k-means. With this visual word representation, three-level spatial histograms are computed on grids of $1 \times 1$, $2 \times 2$ and $4 \times 4$. Histogram intersection (Felzenszwalb et˜al., 2008) is used to define the similarity of two histograms at the same pyramid level for two images. The kernel matrices at the three levels are normalized by their respective means, and linearly combined together using equal weights.

**Spatial Pyramid of Dense SIFT** As with HOG2x2, SIFT descriptors (Lowe, 2004) are densely extracted (Schmid et˜al., 2006) using a flat rather than Gaussian window at 9 scales on a regular grid at steps of 5 pixels: $5 * (1.2)^i$ for $i = 0to9$. The three descriptors are stacked together for each HSV color channels, and quantized into 1024 visual words by k-means, and spatial pyramid histograms are used as kernels (Schmid et˜al., 2006).

### 5.2.3 Classifiers

We used a SVM classifier with spatial HOG and spatial pyramid of dense SIFT as feature for scene classification. The specificity of SVM classifiers is their ability to maximize the margin between classes. Experimentation has been made using the histogram intersection kernel (or min kernel).

**Intersection (Min) Kernel** Histogram Intersection kernel between histograms a, b

$$K(a,b) = \Sigma_{i=1}^{n} min(a_i, b_i), a_i \geq 0, b_i \geq 0$$

Histogramm Intersection Kernel SVM

$$h(x) = \Sigma_{j=1}^{\sharp SV}(\alpha^j \Sigma_{i=1}^{\sharp dim} min(x_i, x_i^j)) + b$$

Complexity:

$$\sharp\text{support vector} \times \sharp\text{feature dimensions}$$

Indeed, straightforward classification using kernelized SVMs requires evaluating the kernel for a test vector and each of the support vectors. For a class of kernels Maji et˜al. (2008) showed that one can do this much more efficiently. In

particular they showed that one can build histogram intersection kernel SVMs (IKSVMs) with runtime complexity of the classifier logarithmic in the number of support vectors as opposed to linear for the standard approach. The trick is to sort the support vector values in each coordinate, and pre-compute. To evaluate, one has to find position of $x_i$ in the sorted support vector values (cost: log # sv) look up values, multiply & add.

$$\begin{aligned} h(x) &= \Sigma_{j=1}^{\sharp SV}(\alpha^j \Sigma_{i=1}^{\sharp dim} min(x_i, x_i^j)) + b \\ &= \Sigma_{x_i^j < x_i} \alpha^j x_i^j + (\Sigma_{x_i^j \geq x_i} \alpha^j) x_i \end{aligned}$$

Complexity:

$$log(\sharp\text{support vector}) \times \sharp\text{feature dimensions}$$

Maji et˜al. (2008) also showed that by pre-computing auxiliary tables we can construct an approximate classifier with constant runtime and space requirements, independent of the number of support vectors, with negligible loss in classification accuracy on various tasks.

Complexity:

$$\text{constant} \times \sharp\text{feature dimensions}$$

See Maji et˜al. (2008) for more details.

#### 5.2.4 Results for Scene Classification

**image-net** Figure˜7 and Figure˜8 show best correct matches, worst correct matches and wrong matches with best confidence value for every 7 classes of our imageNet dataset (The classes are Beach, Country_house, Garden, Moutain, Road, Space station, Street). Each class contains approximately 800 images fetched from imageNet. Figure˜6 shows the influence of the number of negative samples on the classification results. On this dataset our SVM classifier using spatial pyramid of color sift and the intersection kernel obtained 0.82 of mean normalized accuracy, mean average precision was 0.7, and the mean average roc (the mean area under the ROC curve) was 0.92 (see Figure˜9).

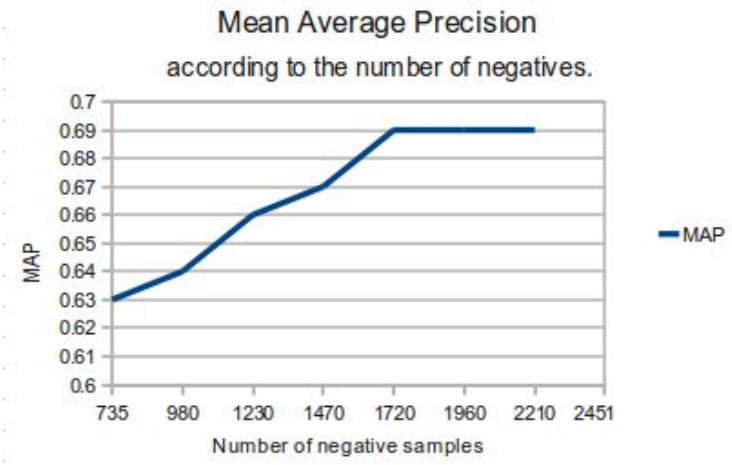



Figure 6: Importance of the number of negative samples in the training set.

| CV/Class | Beach | Country_house | Garden | Moutain |
|---|---|---|---|---|
| Best/Right |  |  |  |  |
| Worst/Right |  |  |  |  |
| Best/Wrong |  |  |  |  |
| Correct was | space_station | Road | space_station | space_station |

Figure 7: Classification results for 7 imageNet classes

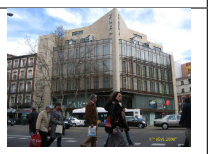
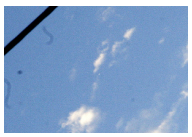
| CV/Class | Road | Space_station | Street |
|---|---|---|---|
| Best/Right |  |  |  |
| Worst/Right |  |  |  |
| Best/Wrong |  |  |  |
| Correct was | street | street | country_house |

Figure 8: Classification results for 7 imageNet classes

| | |
|---|---|
| Mean Normalized Accuracy: | 0.82 |
| Mean Average Precision: | 0.7 |
| Mean Average ROC: | 0.92 |

Figure 9: Classification Accuracy for 7 classes from imagenet

**SUN** Evaluation of a set of classifier mapping 13 classes in the SUN database. These classes have been automatically retrieved from the synopsis of the movie Big Fish (concept extraction + filtering). Figure˜10 and Figure˜11 show best correct matches, worst correct matches and wrong matches with best confidence value for every the 13 classes of the Big fish dataset. Figure˜13 gives classification results for each classes. In this figure, Acc stands for Accuracy, NAcc for normalized accuracy (matches are weighted by the proportion of their class (positive vs negative)), Prec for precision, Rec for recall, AP for average precision and finally aroc for average ROC (area under the ROC curve). As outlined in Figure˜14, mean average precision for this set of classifiers is 0.57.

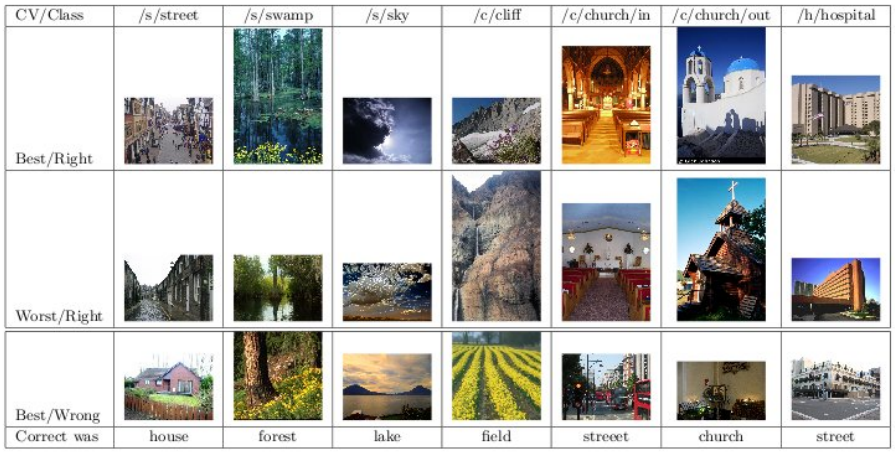



Figure 10: Classification results for the 13 SUN classes

| CV/Class | /h/house | /f/forest | /f/field | /p/pond | /p/plaza | /l/lake/natural |
|---|---|---|---|---|---|---|
| Best/Right |  |  |  |  |  |  |
| Worst/Right |  |  |  |  |  |  |
| Best/Wrong |  |  |  |  |  |  |
| Correct was | street | swamp | pond | lake | street | pond |

Figure 11: Classification results for the 13 SUN classes

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 29 | 0 | 0 | 1 | 6 | 24 | 4 | 46 | 0 | 0 | 1 | 22 | 5 |
| 0 | 22 | 1 | 0 | 0 | 4 | 0 | 2 | 5 | 0 | 6 | 0 | 3 |
| 0 | 0 | 37 | 0 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 1 | 6 |
| 0 | 1 | 0 | 59 | 0 | 2 | 0 | 2 | 0 | 1 | 3 | 1 | 2 |
| 1 | 4 | 0 | 0 | 45 | 14 | 1 | 2 | 0 | 0 | 0 | 3 | 1 |
| 3 | 0 | 0 | 2 | 3 | 88 | 3 | 47 | 0 | 0 | 3 | 8 | 6 |
| 2 | 0 | 0 | 0 | 2 | 4 | 5 | 19 | 0 | 1 | 0 | 4 | 0 |
| 6 | 0 | 0 | 1 | 1 | 31 | 4 | 221 | 2 | 4 | 1 | 14 | 2 |
| 0 | 6 | 0 | 3 | 0 | 3 | 0 | 0 | 32 | 2 | 3 | 0 | 1 |
| 0 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 1 | 62 | 5 | 0 | 14 |
| 0 | 3 | 0 | 2 | 1 | 1 | 0 | 1 | 1 | 9 | 20 | 0 | 16 |
| 5 | 0 | 0 | 1 | 1 | 18 | 0 | 11 | 0 | 1 | 0 | 30 | 2 |
| 0 | 3 | 1 | 7 | 0 | 3 | 0 | 4 | 0 | 6 | 25 | 1 | 80 |

Figure 12: Confusion matrix for the 13 SUN classes

| class | Acc | NAcc | Prec | Rec | AP | aroc |
|---|---|---|---|---|---|---|
| /s/street : | 0.9 | 0.62 | 0.61 | 0.26 | 0.31 | 0.7 |
| /s/swamp : | 0.96 | 0.81 | 0.46 | 0.65 | 0.53 | 0.97 |
| /s/sky : | 0.99 | 0.9 | 0.95 | 0.8 | 0.87 | 0.99 |
| /c/cliff : | 0.96 | 0.9 | 0.58 | 0.85 | 0.76 | 0.98 |
| /c/church/indoor : | 0.96 | 0.88 | 0.64 | 0.79 | 0.79 | 0.99 |
| /c/church/outdoor : | 0.79 | 0.73 | 0.34 | 0.65 | 0.41 | 0.81 |
| /h/hospital : | 0.95 | 0.61 | 0.19 | 0.24 | 0.17 | 0.89 |
| /h/house : | 0.78 | 0.81 | 0.51 | 0.86 | 0.69 | 0.89 |
| /f/forest : | 0.97 | 0.87 | 0.63 | 0.76 | 0.7 | 0.98 |
| /f/field : | 0.96 | 0.89 | 0.69 | 0.81 | 0.77 | 0.97 |
| /p/pond : | 0.9 | 0.69 | 0.21 | 0.46 | 0.19 | 0.82 |
| /p/plaza : | 0.89 | 0.74 | 0.27 | 0.57 | 0.33 | 0.87 |
| /l/lake/natural : | 0.9 | 0.83 | 0.5 | 0.75 | 0.61 | 0.93 |

Figure 13: Classification Results on SUN Dataset

| | |
|---|---|
| Mean Accuracy: | 0.92 |
| Mean Normalized Accuracy: | 0.79 |
| Mean Average Precision: | 0.57 |

Figure 14: Classification accuracy for the 13 SUN classes

## 5.3 Alignment

We are not currently able to shows alignment results for synopsis to video alignment due to technical difficulties at the end of the internship, but our alignment code is available and testes could be performed in a short delay. Figure˜15 illustrates an alignment between concepts extracted from the text and the list of classified movie scenes. Figure˜16 show the desired result of the alignment procedure.

| Match Matrix | beach | street | car | ... | forest |
|---|---|---|---|---|---|
| Once upon a time ———— | 0 | 0 | 0 | 0 | 0 |
| ———— beach ———— | 1 | 0 | 0 | 0 | 0 |
| ——— street ——————— | 0 | 0 | 1 | 0 | 0 |
| ———————— car ———— | 0 | 1 | 0 | 0 | 0 |
| ——— forest ———. | 0 | 0 | 0 | 0 | 1 |

Figure 15: Example of alignment between concepts of a text and classified movie scenes


Figure 16: Scene Description

# 6 Conclusion

During this project, we investigate on three completely different fields: text processing, vision, and sequence alignment. In order to compute an alignment between two texts, we developed a fast (x100) C++ front-end for WordNet 3.0 Jardonnet (2010). This library provides the possibility to evaluate the semantic similarity of two words. This functionality has been used for fuzzy sequence alignment where both texts can refer to same things without using identical words. A consequent part of the project (not presented here) was also focused on bibliographic research on sequence structure understanding, and co-reference resolution (we recommend readers to take a look at Kahane et˜al. (2009); Surdeanu and Manning (2010); Agirre et˜al. (2009) ).
We developed a state of the art scene classifiers using SVM with min kernel and pyramids of SIFT histograms. The set of classifiers is automatically generated from synopsis an analysis, allowing us to train useful classifiers only. These classifiers are then used in order to compute a sequence of classifier scene in a movie.
Finally, few has been said about the alignment procedure. Automatic text to text alignment is operational and qualitative hand-made evaluation (dozens of matches checked for few synopsis to screenplay alignments) revealed that the process seemed to perform quite well. The code for synopsis concepts alignment to movie scenes is ready but, due to a lake of time, we cannot present you quantitative evaluation at this time. However 100% of the code is available and an evaluation could be obtain soon.

# References


Agirre, E., Chang, A.˜X., Jurafsky, D.˜S., Manning, C.˜D., Spitkovsky, V.˜I., and Yeh, E. (2009). Stanford-ubc at tac-kbp. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA. 6

Chen, J., Bangalore, S., and Vijay-Shanker, K. (1999). New models for improving supertag disambiguation. In *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 188–195. 4.2.2

Cour, T., Jordan, C., Miltsakaki, E., and Taskar, B. (2008). Movie/script: Alignment and parsing of video and text transcription. (document), 1

Cour, T., Jordan, C., and Taskar, B. (2009). Learning from ambiguously labeled images. (document), 1

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*. 2.3

Dijkstra, E.˜W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271. 4.2.3

Everingham, M., Sivic, J., and Zisserman, A. (2006). "Hello! My name is... Buffy" automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*. (document), 1

Fellbaum, C., editor (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London. 2.2

Felzenszwalb, P., Mcallester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage, Alaska, June 2008.* 5.2.2

Finkel, J., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages pp. 363–370. 4.2.2

Jardonnet, U. (2010). Wncpp, in-memory access to wordnet in c++. Mail me for more information about the incoming release. 2.2, 6

Kahane, S., Lareau, F., Lattice, U.˜P., and Paris, U. (2009). Abstract meaning-text unification grammar: modularity and polarization. 6

Krizhanovsky, A.˜A. and Lin, F. (2009). Related terms search based on wordnet / wiktionary and its application in ontology matching. *CoRR*, abs/0907.2209. 2.2

Laptev, I., Marszaek, M., Schmid, C., Rozenfeld, B., Rennes, I., Grenoble, I.˜I., and Ljk, L. (2008). Learning realistic human actions from movies. In *CVPR.* (document), 1, 1

Lowe, D.˜G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91. 5.2.2

Maji, S., Berg, A.˜C., and Malik, J. (2008). Classification using intersection kernel support vector machines is efficient. 5.2.3

Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. *IEEE Conf. Computer Vision and Pattern Recog.* 2.3

Nadeau, D., Turney, P., and Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. pages 266–277. 4.2.2

Ney, H. (1992). A comparative study of two search strategies for connected word recognition: Dynamic programming and heuristic search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(5):586–595. 3

Ramabhadran, S., Ratnasamy, S., Hellerstein, J.˜M., and Shenker, S. (2004). Prefix hash tree: An indexing data structure over distributed hash tables. Technical report. 4.1, 4.2.2

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *CoNLL*. 4.2.2

Sankar, P., Jawahar, C.˜V., and Zisserman, A. (2009). Subtitle-free movie to script alignment. In *Proceedings of the British Machine Vision Conference*. (document)

Sankar, P., P, S., and Jawahar, C.˜V. (2006). Text driven temporal segmentation of cricket videos. (document)

Schmid, C., Ponce, J., and Lazebnik, S. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *In CVPR*, pages 2169–2178. 5.2.2

Siek, J.˜G., Lee, L.-Q., and Lumsdaine, A. (2001). *The Boost Graph Library: User Guide and Reference Manual (C++ In-Depth Series)*. Addison-Wesley Professional. 4.2.3

Surdeanu, M. and Manning, C.˜D. (2010). Ensemble models for dependency parsing: Cheap and good? In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference (NAACL-2010)*, Los Angeles, CA. 6

Xiao, J., Hays, J., Ehinger, K., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene regcognition from abbey to zoo. *MIT Press*. 2.3

Zesch, T., Müller, C., and Gurevych, I. (2008a). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*. 2.2

Zesch, T., Müller, C., and Gurevych, I. (2008b). Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI*, pages 861–867. 2.2

# A Appendix

## A.1 Automatic Scene Detection in Text

# 1 Synopsis

The movie begins with Will Bloom 's ( Billy Crudup ) narrative of his father . We see his dad , Edward Bloom , as an old man ( Albert Finney ) fishing . He then turns around and is young Edward ( Ewan McGregor ) . Young Edward has grabbed the biggest catfish ever known to man . He opens the fish 's mouth and grabs his wedding ring . He lets the fish go . Will explains that his dad thinks the big is the spirit some old pirate who is obsessed with gold , therefore his dad attracts the fish with his ring . Now Will is a little boy out camping with some friends . His dad is telling the boys stories . All listen intently except for one , Will . He 's heard his dad 's stories so many times that he 's already sick of them by age 9 . Next Will 's going to his prom , but his date is stuck in the living room[('living_room.n.01', 0)] listening to his dad 's stories . Now Will is having the most important day of his life , his wedding day . But his dad steals the show with more stories . Will ca n't bare it so he tells his new bride , Josephine , he 's stepping out . This is the last time Will spoke to his dad again . Will 's mom , Sandra ( Jessica Lange ) however keeps in touch with him . She always makes excuses for why either Will wo n't talk to his dad Edward wo n't talk to his son . She 's been in the middle of them for years until Will gets a letter from her oneday saying his father is dying . Josephine , who is now pregnant , wants Will to see his father and to come along . Will finally agrees . He comes home to see his dad extremely altered . His father is known for swimming everyday , but the pool[('pond.n.01', 0)] is filled with leaves and unused in months . Soon his dad starts breaking out the stories . It seems Josephine is the only one willing to listen , though she 's heard his stories already as well . Soon Will 's recalling his dad 's first lie.....um , i mean story . Edward 's mother is in labor with him . The doctor tells her to push and out pops baby Edward . He goes sliding down the hospital[('hospital.n.01', 0)] hallway[('corridor.n.01', 1)] . No one his able to catch him . Just as he 's about to hit a wall head first a nurse finally grabs him . Young Edward is in the swamps[('swamp.n.01', 0)] of his hometown with four friends . They encounter a witch 's home . All are afraid to enter , but Edward . Slowly Edward walks toward the spooky house[('house.n.01', 0)] . Suddenly the witch ( Helena Bonham Carter ) opens the door[('doorway.n.01', 0)] . Edward politely tells her his name and that his friends wish to see her eye . ( This eye will tell them how they 'll die ) . Edward goes back to his friends , they ask if he has the eye and he says yes . Then the witch jumps out from behind Edward . Two of his friends have already ran away and two are left . One looks in the eye and sees he 'll die from falling off a ladder as a very old man . Another sees he 'll die as a young man on a toilet from a heart attack . Now Edward looks and says , " Oh , so that 's how I 'll die " . Edward 's in church[('church.n.01', 0)] singing , suddenly his voice is changing rapidly and his feet are growing . Edward has some growth disorder and has to lay in bed for months while in some device . The whole time he 's in bed he reads the encyclopedia . Edward has now grown into a strapping young man and the pride and joy of his hometown . He wins a championship basketball game for the town . Everyone carries him in the air

with joy . Everyone except his friend from the woods[('forest.n.01', 0)] who saw himself die as a young man , we 'll just call him Jimmy Smith . Jimmy sits with a jealous look on his face towards Edward . Edward wins a football game . Jimmy shakes his head in disbelieve . Edward saves a cat from a burning house[('house.n.01', 0)] . Everyone rejoices.....except Jimmy . Soon peoples sheep , chickens , and dogs are being eaten . There 's word of a giant among the people . Everyone decides someone should go get rid of it . Edward volunteers himself . Edward goes into the woods[('forest.n.01', 0)] and finds Carl the giant . Once Edward sees Carl emerge from his cave he knows he 's done for . So he tells Carl he 's been sent as a human sacrifice . ( therefore basicly saying , " I give up . " ) Carl tells Edward he does n't wan na eat him , and that he 's just always so hungry . Carl seems sad about this . Edward then tells Carl he 's just a big fish in a little pond[('pond.n.01', 0)] and encourages him to leave the little town for bigger things . Edward realizes it 's time he does the same . He then leaves with Carl . The town is sad to see Edward and Carl , whom they 're no longer afraid of , leave . And throw them a going away celebration . Edward and Carl walk along the path to their future when Edward sees a sign . This sign leads down another path through the woods[('forest.n.01', 0)] . This path Edward has always wanted to follow since he was little . He tells Carl to continue the path they 're following and he 'll go through this mystery passage . Carl thinks Edward 's trying to ditch him , however Edward promises he 'll meet him on the other side . Edward walks through the very difficult path . There 's jumping spiders and bees that attack him , but he feels there 's always obstacles to face when something great is waiting on the other end . Edward finally comes across a little town called Spectra . The road is paved with grass and everyone is barefoot . It 's the pleasantest little town Edward has ever seen . The mayor of Spectra tells Edward he 's too early . Edward sees his name on the mayor 's list . They want him to stay anyway . Edward meets the town folks including a poet from his very own hometown , Norther Winslow ( Steve Buscemi ) . Edward always thought Norther was somewhere in France , but it turns out he 's been in Spectra all along . Soon Edward wants to leave Spectra . Little Jenny , a girl Edward has met there , does n't want him to go . It appears she has a crush on him . She begs him to come back someday and he says he will . Jenny had earlier taken Edward 's shoes so the trip back through the path is very painful for him . Edward comes back to the path Carl took , and to his surprise Carl is still there waiting ! Edward is at a circus[('stadium.n.01', 1)] . There 's a big hoopla over this " giant " throughtout the audience . Edward seems the least bit phased . He whistles to the lighting guy and points next to him . Everyone sees Carl . The ringmaster , Amos Calloway ( Danny DeVito ) is in love . He tells everyone the shows over as he approaches Carl . Edward now sees a beautiful young woman . He says time stops when you 've found the person you know you 'll marry and so time stops . Edward approaches the girl and time he reaches her time starts back in turbo speed and she 's gone . Carl signs up with the circus[('stadium.n.01', 1)] and Edward begs for a job as well . Amos finally gives in when Edward agrees to work for free only to be told each month something about is dream girl whom Amos knows . Month one : Edward

is giving an obeised man a bath[('bathroom.n.01', 0)] and Amos tells him his girl likes daffodils . Edward is on cloud nine repeating over and over again , " Daffodils , she likes daffodils " . Month two : Amos tells Edward his girl goes to college . Edward is in a ring full of motorcyclist jumping over his head , but he pays them no mind as he keeps saying , " College , she goes to college " . Month three : Edward is in a cannon that shoots him in the air , but he 's still daydreaming about his girl . Finally one night Edward discovers Amos turns into a wolf at night . Edward befriends the wolf , not knowing it 's Amos , and Amos respects him the next morning for taking care of him when others would have been afraid . He then tells him is girls name is Sandra ( Allison Lohman ) . Edward goes to Sandra 's college . She comes out and he tells her he wants to marry her . Though flattered Sandra is already engaged . Edward walks away saying any sensible man would give up . We then see him running back and he says he 's no sensible man . Edward 's in a field[('field.n.01', 0)] of daffodils and he screams out to Sandra that he 's gon na marry her . Her fiancee comes , it 's Jimmy Smith from back home ! Sandra begs Edward not to fight him so he does n't . He instead gets the crap beat outta him by Jimmy . Finally Sandra tells Jimmy she 'd rather marry a complete stranger than him . She sits next to a badly bruised Edward and he smiles at her with some teeth missing . Jimmy 's on the toilet reading a girly magazine when he has a heart attack and dies . ( remember , the witch 's eye foretold this ) Edward is in the hospital[('hospital.n.01', 0)] recovering when he finds out he 'll be drafted into the war . Before he goes he marries Sandra . While serving in the army Edward decides to take the most risky assignments , hoping to get an early leave to get home to Sandra . For one of these assignments Edward must retrieve some top[('tent.n.01', 2)] secret files from the Vietnamese army . Upon accepting the job Edward has to sky[('sky.n.01', 0)] dive to the target area . In the area numerous Vietnamese soldiers are watching a show given by two conjoined twins , Ping[('river.n.01', 1)] and Jing . Edward drops[('cliff.n.01', 0)] backstage and gets the files after fighting many soldiers . However when the curtain drops[('cliff.n.01', 0)] after the show everyone sees Edward 's parachute and know he 's there . Edward 's trapped and the twins find him in his hiding place[('plaza.n.01', 0)] . He then begs the twins to help him . They 're moved by his story of his having to leave Sandra , plus he offers them a job in Amos Calloway 's circus[('stadium.n.01', 1)] . Back at home Sandra receives word that Edward is dead . She 's heart broken . Months pass and Edward surprises Sandra , she ca n't believe her eyes . Edward is older and he 's a travelling salesman . He goes to the bank one day and sees Norther Winslow , who has finally left Spectra . Norther informs him he 's there to rob the bank . He makes Edward an accomplice and tells him to get the rest of the money out of the vault[('burial_chamber.n.01', 1)] . The cashier informs Edward nothing is there . While in the get away car Edward tells Norther that he could n't get anything out of the vault[('burial_chamber.n.01', 1)] because it had nothing in it . They go their separate ways and Norther comes to the conclusion that he 's going to Wall street[('street.n.01', 0)] where the big money is at . Now we 're back with Will . He 's cleaning out his dad 's office[('office.n.01', 0)] . He

sees a document about Jenny from Spectra . He thinks his dad has another family with her . Will then goes to Spectra himself . Once in Spectra Will meets Jenny who is older ( Helena Bonham Carter ) . She tells Will the story of his dad 's second visit to the town . One rainy night Edward can barely see the road as he travels home . Soon he 's in water . He sees a naked lady that he remembers seeing back in Spectra . He gets back on dry land that morning and sees he 's indeed back at Spectra ! The last time he came he was too early , this time he 's too late . Spectra is run down . The grass paved road is gravel . Soon Edward decides to buy the town with the help of Norther Winslow , who is now a millionaire . He owns everything except for a little raggedy house[('house.n.01', 0)] . He finds out Jenny owns the house[('house.n.01', 0)] and she refuses to sell it to him . With the help of Carl the giant Edward fixes up her home and she finally agrees . She 's still in love with Edward and tries to kiss him . He tells her of his wife and stops her . She never sees Edward again . Jenny tells Will that the house[('house.n.01', 0)] become ruined again and she become known as a witch . Will goes back home and his father is in the hospital[('hospital.n.01', 0)] . His dad tells him this is how he dies and he begins to panic . He tells Will to finish the story of how he dies . Will ca n't because his dad has never told him that story . Suddenly Will takes over his dad 's storytelling and begins telling the story of how his dad will die . He 'll take his dad out of the hospital[('hospital.n.01', 0)] and after that escape he 'll take him to the lake[('lake.n.01', 0)] . All he friends are there to see him off , Carl the giant , Ping[('river.n.01', 1)] and Jing , Amos Calloway , Norther Winslow , all of them . Edward will goodbye to them and Will places[('plaza.n.01', 0)] him in the water . He turns into a big catfish and swims off to live on for eternity . ( We 're now back to reality ) Edward smiles at Will and says that 's exactly how it happens and dies . At Edward 's funeral Will sees Carl the Giant for the first time . He 's tall , but not exactly as tall as his dad described . We also see the conjoined twins . They are twins , however not conjoined . We see Jenny and Norther Winslow too . Everyone is gathering and telling Edward 's stories . Will says it 's like hearing a joke you have n't heard in a while and suddenly it 's funny to you again . He also says his dad knew what he was doing . He knew that by telling his stories he 'd continue to live on for generations .

---

## 2 Concepts

hospital.n.01
pond.n.01
street.n.01
doorway.n.01
cliff.n.01
corridor.n.01
office.n.01
field.n.01
plaza.n.01

tent.n.01
wave.n.01
church.n.01
burial_chamber.n.01
lake.n.01
river.n.01
living_room.n.01
swamp.n.01
stadium.n.01
house.n.01
operating_room.n.01
forest.n.01
bathroom.n.01
sky.n.01