# Multimodal Alignment of Scholarly Documents and Their Presentations

**Bamdad Bahrani**

**(B.Eng, Amirkabir University of Technology)**

Submitted in partial fulfillment of the

requirements for the degree

of Master of Science

in the School of Computing

**NATIONAL UNIVERSITY OF SINGAPORE**

2013

# Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Bamdad Bahrani
03/28/2013

*To my parents, without whom, it was not possible for me to improve...*

# Acknowledgments

I would like to thank my supervisor Dr. Kan Min-Yen for his invaluable guidance through the rout of my graduate education.

# Contents

# List of Figures

# List of Tables

**Abstract**

We present a multimodal system for aligning scholarly documents to corresponding presentations in a fine-grained manner (i.e., per presentation slide and per paper section). Our method improves upon a state-of-the-art baseline that employs only textual similarity. Based on an analysis of errors made by the baseline, we propose a three-pronged alignment system that combines textual, image, and ordering information to establish alignment. Our results show a statistically significant improvement of 25%. Our result confirms the importance of emphasizing on visual content to improve document alignment accuracy.

# Chapter 1

# Introduction

Scholars use publications to disseminate scientific results. In many fields, scholars also congregate at annual congresses to narrate their scientific discoveries through presentations. These two vehicles that document scientific findings are interesting in their complementarity; while they overlap in content, presentations are often aimed at an introductory level and may motivate one to take up the details in the more complete publication format.

As the presentation is often more visual and narrated by an expert, it can be regarded as a summary of the salient points of a work, taken from the vantage point of the presenter. By itself, certain presentations may fulfill information needs that do not require in-depth details or call for a non-technical perspective of the work (for laymen as opposed to subject matter experts). It is thus clear that a useful function would be to link and present the two media – scholarly document and presentation slides – in a fine-grained manner that would allow seamless navigation between both forms. In this thesis, we further the state of the art towards achieving this goal, by designing and implementing a multimodal system that achieves such functionality.

## 1.1 Motivation

There have been tens of millions of papers published in the academic world since 1750 (Jinha, 2010). Although many are accessible only in hard copy, more than 2/3rds exist in a digital format, found in electronic libraries and online databases. Most recent published work – 1990 to present – are in electronic forms, of which Portable Document Format (PDF) is the current predominant format. PDF is now an open standard, and is readable through software libraries for most major computing and mobile device platforms.

Scientists disseminate their research finding in both written documents and often in other complementary forms such as slide presentations. Each of these forms of media has a particular focus, and as such, while some of the information may be redundant, some is unique to a particular media form. A key differences between these two forms of knowledge transportation is the detail level. Papers are often more detailed than presentations since they are a comprehensive archival version of research findings. Scientific papers often formalize the problem and explain the solution in depth, covering the minutiae and complexities of their research, if any. In contrast, slide presentations largely omit details due to their nature: as they usually narrated in a time-limited period, they are often shallow, and describe the scholarly work at a high level, using easy-to-understand arguments and examples. In other words, papers and presentations serve two levels of seeking knowledge: paper format yields deeper technical knowledge needed to implement or reproduce a study; whereas presentation is the shallow level which users may only need to browse the outline of the research. As slide presentation are a more shallow form of knowledge representation, scholars have also viewed them as a well-structured summary of the deeper paper form. Often times, the presentation originates from the same author and describes the key issues of the paper. Reading this summary, one may seek more information by reviewing the slides in detail or read the respective

sections of the paper.

These statements support the need for simultaneously reading through both paper and presentation together. Such a facility would be useful to users who need to review a study in two level of details simultaneously.

In this research, we design and implement a system which maps both versions of a same research: a scholarly paper alongside with its slide presentation. The generated map shows the relation between slides of the presentation and sections of the paper. Using this map, readers can switch between the two representations of the research.

## 1.2 Problem Definition

Previous work has addressed finer-grained alignment on paragraphs to slides (Ephraim, 2006; Kan, 2007). These previous works observed that in many cases, the alignment is better characterized as aligning several paragraphs of a document to one slide. Therefore, we define our problem in a way that documents are represented at the granularity of (sub)sections, rather than single paragraphs.

We formalize the problem of *document-to-presentation alignment* as follows:

Given: Presentation $S : s_{1,\ldots,n}$

Document $D : d_{1,\ldots,m}$

Output: Alignment $f(S, D) = AM$

which gives an Alignment Map (AM) of presentation $S$ and document $D$. Each presentation $S$ contains $n$ slides $s_{1,\ldots,n}$ and each paper $D$ contains $m$ sections $d_{1,\ldots,m}$. AM is a $n \times m$ matrix which shows the aligned section for each slide. Each row represents one slide ($s_i$) and determines the respective section of the paper which is aligned to that. The system may also decide the slide $s_i$ should not be aligned to any sections of the paper, defined as a *nil* alignment. Take note that we define the

problem in a way that each section of the paper may be aligned to several slides from presentation, but each slide can only be aligned to maximum one section of the paper. Figure 1.1 schematically shows the problem we try to address.

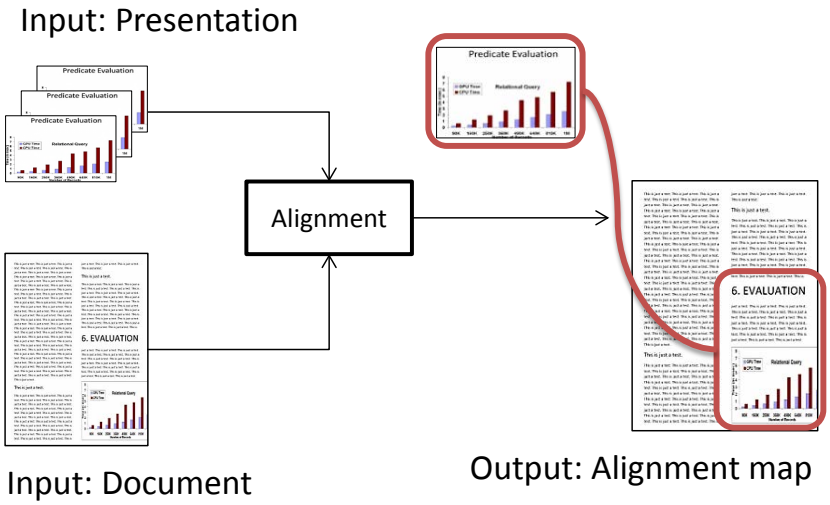


Figure 1.1: Simplified diagram illustrating our problem definition.

## 1.3 Solution

To build a baseline, we first approach this problem from an information retrieval perspective. For each slide $s$ we retrieve the most similar (sub)section from the paper ($d$) and claim that $d$ is the most probable section to be aligned to slide $s$, following the assumption also made in previous work (Beamer and Girju, 2009; Ephraim, 2006; Hayama, Nanba, and Kunifuji, 2005; Kan, 2007). None of these previous works, however, have taken advantage of the inherently visual content in slides as evidence for alignment. Our work rectifies this shortcoming: our multimodal system benefits from both textual content and the visual appearance of slides to generate its alignment. Although some previous studies (Hayama, Nanba, and Kunifuji, 2005) suggests that slides formatting can be leveraged, to our best of

knowledge, our work is the first to actually employ visual information in the alignment process. Our system also retains the best practices from previous work by preferring 1) (partial) monotonic alignments and 2) catering for *nil* alignments. By monotonic alignment, we mean that our system prefers to align slides to follow the same flow as the paper sections. By *nil* alignments, we mean slides which should not be aligned to any paper sections.

## 1.4 Organization

This thesis has six chapters. Chapter 2 reviews related work in presentation processing and generation, text similarity and alignment, and synthetic image classification. In Chapter 3, we conduct an analysis of our slide dataset. Chapter 4 presents the core contribution of this thesis: the methodology used in our multimodal alignment. We review the system components including preprocessing, text alignment, image classification and late fusion units. A key aspect of our work is the novel incorporation of an image classifier, so we describe this component and its evaluation in detail. In Chapter 5, we evaluate our alignment system and conclude the thesis in Chapter 6.

# Chapter 2

# Related Work

We now relate how previous and background work informs our thesis. We examine prior in three related fields. 2.1 discusses presentation processing: slide and presentation retrieval, presentation generation as well as presentation-to-paper alignment.

Since our system is multimodal, we also review both text and (synthetic) image processing pertinent to our method, in the two separate sections following.

## 2.1 Presentation Processing

Studies on presentation processing range in topic from slide retrieval and reuse to presentation generation and presentation to paper alignment.

A few studies show the importance of proper slide structure identification: *i.e.* differentiation between presentation body and title text, identification of graphical elements such as figures, charts and plots. Such structure is leveraged in downstream applications, e.g., in slide reuse. In (Hayama, Nanba, and Kunifuji, 2008), a method is proposed to extract visual structure underlying a presentation to facilitate the reuse of the content of existing presentations. They used textual attribute information as well as visual cues on the slides to detect structure of the presenta-

tion slides. Presentation structure is also exploited in slide information retrieval. In (Liew and Kan, 2008), when a query is made, a hybrid approach retrieves using both text and image content as evidence. The authors dissect slide images into visually coherent parts, and order the retrieval of the parts according to their relevance to the query. Later (Hayama and Kunifuji, 2011), identify the relationships between the content components to improve slide retrieval performance.

Another application of structure identification is presentation generation from documents, that work either in a fully-automated (Shibata and Kurohashi, 2005; Sravanthi, Chowdary, and Kumar, 2009) or semi-automated approaches (Gokul Prasad et al., 2009; Hasegawa, Tanida, and Kashihara, 2011; Wang and Sumiya, 2012). In (Shibata and Kurohashi, 2005), an automatic procedure is introduced that can generate slides by processing raw text. It takes advantage of syntactic analysis to identify units such as sentences and clauses and the relationship among them in Japanese. Then it distinguishes topic and non-topic parts and arranges them in the presentation according to syntactic units. While some automatic generation techniques are suited for raw text, others are only applicable for papers with standard formats. (Sravanthi, Chowdary, and Kumar, 2009) rely on popular proceeding and journal template formats to generate slides; the document is first processed and converted to an internal XML representation, which is used to extract key phrases and sections. The identified key phrases are input to a query-base summarizer that generates the slides.

Prior work has also made use of a database of pre-made presentations as a source for generating new ones (Hasegawa, Tanida, and Kashihara, 2011; Wang and Sumiya, 2012): Hasegawa *et al.* (Hasegawa, Tanida, and Kashihara, 2011) propose a framework that assists amateurs to assemble presentation by applying heuristics. In (Wang and Sumiya, 2012), the relationship between the words in previous pairs of text and presentation is derived which describes the relationship between the

way each word is expressed in text and its corresponding presentation. The same style is then extended to new presentations.

We discussed several studies on automatic generation of slide presentations from academic papers so far. Most of them need to apply machine learning techniques on many pairs of scientific papers and presentations. (Hayama, Nanba, and Kunifuji, 2005) and (Beamer and Girju, 2009) suggest that the first step on this route is to present a method for aligning papers and presentations together in a fine-grained level. Hayama *et al.* (Hayama, Nanba, and Kunifuji, 2005) first tackled this problem with Japanese technical papers and presentation sheets using a Hidden Markov Model suggested by Jing (Jing, 2002). The idea behind Jing's HMM, in this context, is to find the most likely position in the paper for each word that appears in the corresponding presentation by exploiting a combination of heuristic rules. According to these rules, the probability that two adjacent words in a presentation slide refer to two adjacent words in a particular sentence is higher than them referring to two words in different sentences or even two non-adjacent words of the same sentence. The transition probability between position of adjacent slide's words is determined two by two based on these rules. At last the word sequence with the highest probability is derived as the final result.

The idea of aligning presentations and papers was then taken up by Kan (Kan, 2007) with the SlideSeer digital library, which enlarged the scope of the alignment work to include the crawling of document-presentation pairs and bimodal browsing (presentation- or document-centric) user interface. Claiming that more complex algorithms failed to increase alignment accuracy in (Kan, 2007), Kan uses maximum similarity as his baseline method for aligning. Maximum similarity is a greedy model which simply aligns a target slide to the paragraph with the maximum textual similarity. He uses a paragraph spanning algorithm to gain more exact results. More recently, Beamer and Girju (Beamer and Girju, 2009) performed

a detailed analysis of different similarity metrics' fitness for the alignment. Their evaluation results show that a scoring method which simply based on the number of matched terms between each slide and section is superior to other methods.

(Beamer and Girju, 2009; Ephraim, 2006; Hayama, Nanba, and Kunifuji, 2005; Kan, 2007) all mention the need of identification of slides that should not be aligned, defining them as *nil* slides. Hayama *et al.* (Hayama, Nanba, and Kunifuji, 2005) eliminate around 10% of their presentation sheets which they assume *nil* and report that this causes 4% of improvement in their final results. Beamer and Girju in (Beamer and Girju, 2009) conclude that if they had a *nil* classifier, they could have gain around 25% higher accuracy in their results. They manually remove the "non-align able" slides and that increases their final accuracy from around 50% to 75%. Kan in (Kan, 2007) structures this challenge as a supervised machine learning problem and tries to classify *nil* slides and mark them as non-aligned. He however, reports that classifying *nil* slides causes a small percentage gain of only 3% in his experiments which he shows is a significant improvement according to his results. Also (Ephraim, 2006) classifies a slide as *nil* when it cannot be aligned to any paragraph and observes performance improvement of 1% to 11%.

Although a lot of research effort has been made to exploit presentation structure for the purpose of slide reuse, retrieval, and presentation generation, there has been minimal work up to now to incorporate this information for document-presentation alignment. Previous studies on this specific task have maintained a text matching approach and were not able to achieve alignment accuracy of more than 63% in their results. An aspect that was found useful in many of the presentation structure extraction studies, but has yet to be leveraged in alignment task is the visual content of the slides.

We contribute to the state-of-the-art by addressing this weakness. Our system builds from existing text similarity baselines (Kan, 2007; Beamer and Girju,

2009), exploiting graphical information to specifically correct weaknesses text-only alignment when dealing with certain classes of presentation slides. In our proposed method an image classifier is designed to distinguish four type of slides according to their visual appearance. The system then applies heuristic rules on different slide classes to improve the text-only alignment results. We detail the proposed system and the alignment pipeline in the upcoming chapters.

## 2.2 Text alignment and Similarity measures

Text alignment looks for equivalent units of text between two or more documents and aligns them to each other. The granularity of the text unit can vary: entire documents, paragraphs, sentences or even individual words. Input documents can be of the same language or translations in different languages. Thus, our alignment task can be cast as an instance of this framework, where the two inputs express information in two different languages. Finding equivalent text units can be seen as a special type of Multilingual Text Alignment (MTA).

Multilingual text alignment is a well-studied research area as it is a pre-requisite to machine translation. MTA methods can be divided in two general classes (Wu, 1994). The first class which relies only on the available textual sources and examples which takes a statistical approach. The second class relies on lexical information, which may be obtained from external knowledge sources. For example, lexical approaches may use an external bilingual lexicon to match textual units. Statistical MTA approaches calculate all possible alignments and chose the one with maximum probability. Although statistical methods rely on little domain knowledge, they generally perform better than more sophisticated lexical approaches (Gale and Church, 1991).

Alignment approaches rely on a core similarity measure, to calculate the similarity between spans of text. To best understand current approaches to this

area, we first review how a text document is represented in vector space (Huang, 2008). Each document consists of words. If we count the frequency of each word occurrence and assume that each word corresponds to a dimension in the resulting data space, then each document becomes a vector consisting of non-negative values on each dimension.

Let $D = \{d_1, ..., d_n\}$ be a set of documents (sections in our case), and $T = \{t_1, ..., t_m\}$ the set of distinct term occurring in $D$. A documents is then represented as an $m$-dimensional vector $\overrightarrow{t_d}$. Let $tf(d, t)$ denote the frequency of term $t \in T$ in document $d \in D$. Then the vector representation of a document d is $\overrightarrow{t_d} = (tf(d, t_1), ..., tf(d, t_m))$

With document presented as vectors, the degree of similarity of two documents can be measured as the correlation between their corresponding vectors (Huang, 2008). There are several methods to measure that i.e. Euclidean distance, Cosine similarity, Jaccard index, Person correlation coefficient and Luccene's similarity. Following is the explanation of some important ones.

Euclidean distance which is the default distance measure used with the K-means algorithm is the ordinary distance between two points that one would measure with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text (Huang, 2008). Computing of Euclidean distance for two documents given their term vectors $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ are the same as computing the distance between two vector.

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Given two documents which are represented by their term vector $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$, cosine similarity is calculated as

$$CosSim(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|} \tag{2.1}$$

The Jaccard index, also known as the Jaccard similarity coefficient is a statis-

tic used for comparing the similarity and diversity of sample sets. It measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms (Huang, 2008):

$$JacSim(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| + |\overrightarrow{t_b}| - \overrightarrow{t_a} \cdot \overrightarrow{t_b}} \tag{2.2}$$

For documents to be shown as vectors, counting the number of occurrences is not the only way. Instead, the weights of the terms, or the importance of them can be computed as used to represent document vector. Term frequency, inverse document frequency ($tf.idf$) is a numerical statistics which reflects the importance of a word to a document, with respect to a collection of documents or corpus. This is a very common way to control the fact that some words are generally more frequent than others and was first introduced by Salton in (Salton, 1984). $tf.idf$ for each word is calculated as the multiplication of its two factors: $tf$ and $idf$. Term frequency for term $t$ on document $d$ ($tf(t,d)$) is the frequency with which term $t$ occurs in document $d$. This value can be normalized by dividing by the number of terms in document or the maximum $tf$ of all of the terms in that document:

$$tf(t,d) = \frac{f(t,d)}{max\{f(w,d) : w \in d\}} \tag{2.3}$$

Inverse document frequency is a measure of whether the term is common or rare across all documents ($D$). It is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of this score:

$$idf(t,D) = \log \frac{|D|}{|\{d \in D : tf(t,d) \neq 0\}|} \tag{2.4}$$

Some studies also suggest different methods for measuring the similarity between short segments of text (i.e search queries, tags, newspaper sentences and its summary) (Metzler, Dumais, and Meek, 2007; Yih and Meek, 2007; Jing, 2002). Looking the alignment problem from and IR approach, (Voorhees, 1994; van der Plas and Tiedemann, 2008) suggest that query expansion tends to help performance with short, incomplete queries but degrades performance with longer, more complete queries. Beamer and Girju in (Beamer and Girju, 2009) take their suggestion and implement such method for the specific problem of aligning paper documents to slide presentations. They conclude that query expansion does not have any significant effect on their alignment result. This can be justified by the fact that both presentation and paper are made by one person –the author– and therefore she/he uses the same terminology in them.

In our study the input unit –slides and sections– are not as short as mentioned studies. We take advantage of cosine similarity utilizing $tf.idf$ for similarity measure as our baseline.

## 2.3 Synthetic Image Classification

A successful classification scheme must ensure that it can classify most items and that items clearly belong to distinct classes (Wang and Kan, 2006). Taking account of this fact, (Swain, Frankel, and Athitsos, 1996) and (Wang and Kan, 2006) divide all images into two categories of natural(photographs) and synthetic(computer generated drawings). Their studies both implement binary classifiers which distinguishes between the two mentioned classes of images. Wang (Fei, 2006) consider this as his first level classification in which he ignores natural images because his system is to analyse and classify synthetic images He then introduces NPIC, a hierarchical approach for classification of synthetic images. (Fei, 2006)'s classification on synthetic images has five broad categories: maps, figures, icons, cartoons and artwork.

These classes is considered as his second level classification. On a hierarchical basis, he then breaks them into lower levels. His classifier divides figure class into seven subclasses including illustrations, tables, block diagram and different type of charts (i.e. bar chart, line chart, pie chart). To our knowledge, few studies have focused specifically on synthetic image classification except (Wang and Kan, 2006; Fei, 2006) and (Lienhart and Hartmann, 2002). Lienhart and Hartmann (Lienhart and Hartmann, 2002) present algorithms for a 3-class classification. They first categorize images into two classes: 1. Photo/Photo-like images, and 2. Graphical images. Within Graphical images – also defined as synthetic images – they define 3 subclasses: 1. Presentation slide/Scientific posters, 2. Comic/Cartoons and 3. Other images. They devote one category for presentation slides alongside with scientific posters and distinguish this subcategory by observing uniform characteristics about this class. In their observation, there are 3 main differences between presentation slides/scientific posters class and comics class: 1. the relative size and/or alignment of text line occurrences, and 2. the (lack of) containment of multiple smaller images which are aligned on a vertical grid, and 3. their width-to-height ratio (slides are generally 4:3). Motivating by these observations, they extracted several image features and achieved 95% of accuracy in this specific classification.

Huang *et al.* introduce a model based system which identifies scientific charts (Huang, Tan, and Leow, 2004) and attempts to recover their data. Their system recognizes charts and recovers the underlying data. It first separates graphics from text. Then, based on the image's vectorization, extracts the lines and arcs from the image. They build a model on these lines and arcs and use this model to predict the likelihood that a new test image fits into four kinds of chart models (Bar chart, Pie chart, Line chart, High-low chart). They observed that in a chart image, the color or greyscale level within a graphical component is consistent. On the other hand, the color difference or greyscale level difference between neighbouring graphical

components is normally significant. In a follow-up work (Huang, 2008), Huang extends their approach beyond lines and arcs to general shape detection, further improving the classification and data recovery from charts in a single pass.

Selecting suitable features is a critical step for successfully implementing image classification(Lu and Weng, 2007). Wang (Fei, 2006) distinguishes two general feature sets in his work: textual features and visual features. Textual feature examples are image file name, detailed information available from the image properties, or the textual context where the image appears. The limiting factor is if you have lots of images with numbers in their file name, with no other metadata, these features can not be very useful.

Visual features are the other feature class. These rely on the image's visual content, giving rise to Content Based Image Retrieval (CBIR). Content-based means that the search will analyze the actual image content, rather than its metadata such as keywords, tags or descriptions associated with the image. The term "content" might refer to colors, shapes, textures, or any other information that can be derived from the image itself. Swain *et al.* (Swain, Frankel, and Athitsos, 1996) introduces an image search engine which relies on both textual and visual features. Most common visual features are based on the images height and width (Lienhart and Hartmann, 2002; Swain, Frankel, and Athitsos, 1996), color histogram, texture, edge shape (Lienhart and Hartmann, 2002), regions (Fei, 2006), gradient (Ye et al., 2005; Dutta et al., 2009) and pixel value.

We take note of recent visual features. Ye *et al.* (Ye et al., 2005) and Dutta *et al.* (Dutta et al., 2009) suggest using image gradients for extracting text from images and video frames. It also has been shown that image gradients are invariant against different color spaces, illumination changes, and affine transformation such as rotation, scaling and translation (Lowe, 1999). While (Lienhart and Hartmann, 2002) tries to distinguish presentation slides from comics and (Huang, Tan, and

Leow, 2004) attempts to classify different charts, they both use the more basic feature of image edges as an important feature.

A recent feature that has not been used in synthetic image classification is the Histogram of Oriented Gradients (HOG). HOGs have been widely applied on challenging vision tasks in which the image can be represented by shape features: object detection (shape of an object e.g. "car") (Zhang, Zelinsky, and Samaras, 2007), pedestrian detection (vertical structure of human body) (Dalal and Triggs, 2005) and face recognition (face configuration) (Albiol et al., 2008). The problem of slide image classification is similar to the aforementioned problems in the way that the synthetic images of the slides can be represented by shape features characterized by slides elements including background, bars, curves, points, arrows, table elements and general texts.

HOG counts occurrences of gradient orientations in localized portions of an image.The technique relies on both the gradient and edges in an image and is more robust than its predecessor features. Dalal *et al.* (Dalal and Triggs, 2005) suggests that the linear SVM classifier works best with HOG. Also, it has been recently shown that HOG can be applied for text detection/extraction from images (Zhang and Kasturi, 2010). Indeed, HOG improves the gradient features in two phases: 1. pooling and 2. spatial blocks normalization which will be discussed in Image Classification section (4.2)

In this thesis a multimodal alignment system is proposed. Our suggested solution benefits from a combination of textual and visual content of the slides. For text content, we discussed several common text similarity measures in this chapter. For visual content, we investigated (synthetic) image classification schemes. Taking advantage of the background studies that we discussed, the rest of the paper attempts to address the problem in a novel way.

# Chapter 3

# Slide Analysis

To ensure that our approach builds upon the state-of-the-art, we invested effort to study the output and the errors made by a state-of-the-art alignment method. The analysis in this chapter forms the motivating basis for our slide image classification system that is the key component of our multimodal approach to presentation-to-paper alignment. We first describe the dataset used for our study, then describe our slide image categories and how the text-only baseline alignment system fares on aligning slides from these categories.

We take the publicly available document-presentation pair corpus from (Ephraim, 2006) as a starting point. The dataset consists of 20 pairs of papers and presentations. Papers are drawn from DBLP, a metadata repository of computer science papers containing links to the electronic copy (in PDF) when available. For the 20 pairs in this corpus, the papers in PDF form and the presentations in Microsoft PowerPoint format (.ppt) are available and verified to have been constructed by an author of the original paper.

The title, author, and year of the publication were manually cross-checked to ensure data cleanliness and quality. Importantly, the dataset is annotated with ground truth alignments, including annotations of non-alignable slides (*nil*). For

Table 3.1: Demographics from Ephraim's 20-pair dataset.

| | |
|---|---|
| Total # of slides | 751 |
| Average # of slides per presentation | 37.5 |
| Total # of sections | 515 |
| Average # of sections per document | 25.75 |

the purpose of our study, presentations are broken down into an ordered list of individual slides, and papers are broken into an ordered list of sections. We define sections of the paper as a block of paragraphs which have a unique numerical identifier. Basic demographics about the dataset is shown in Table 3.1.

## 3.1 Slide Categorization and Statistics

Through our own observation of the slide images, we have formulated a classification scheme for the types of slides present in the presentations in the corpus. Our classification scheme considers the roles of the text and images. While our classification is based solely on our dataset, we hypothesize that such classes are general for most presentations that place information dissemination as the core objective of its purpose (i.e., as opposed to sales, or collections of wallpapers or illustrated quotations). Taking a slide-centric approach to analysis, we observe the below categories of slides:

- *Nil.* These can be title, example slides, ending slides (Q&A, references) or any other content not directly extracted from the paper. The previous works report that classifying such slides correctly may improve alignment performance anywhere from 3 to 25% (Beamer and Girju, 2009; Kan, 2007).

- **Outline**. These are an important sub-class of *nil* slides, that we have separated from the main class. These slides exist solely to present or recap the presentation structure, to help sync the audience to the material being pre-

sented. Agenda, index or outline slides are some examples same as the slides on Figure 3.1.

- **Image** slides consist almost solely of images. These are challenging to align for the baseline, since there is little or no textual evidence for alignment. Examples of Image slides are shown in Figure 3.2.

- **Table** slides contain tables. Text extraction often extracts the textual strings within the table –stored digitally in the presentation file–, which when extracted verbatim from the document do constitute good evidence for textual alignment. However many Table slides do not have much digitally textual data since the author has used the image of the table instead. Our text extraction process does not utilize any OCR systems to extract text stored in images of the slide.

- **Drawing** slides consist of drawing elements: simple shapes, arrows, graphs and text boxes, authored within the presentation software. The difference between this category of slide and Image slides is that these slides are usually made using the presentation software features. They often include many textboxes which even if their textual content is extracted, there cannot be valuable string for text similarity measures. Some examples of Drawing slides can be seen in Figure 3.3.

- **Text**. Finally, in case that a slide does not contain any major image, table or drawing, it is considered as a Text slide. These slides contain sufficient textual information for us to be able to perform text alignment on them.

Table 3.2 gives the distribution of these slide categories in the dataset by 1) binary presence in the presentations, and 2) by raw number. For each category, the table shows the number of presentation in which there was at least one occurrence of

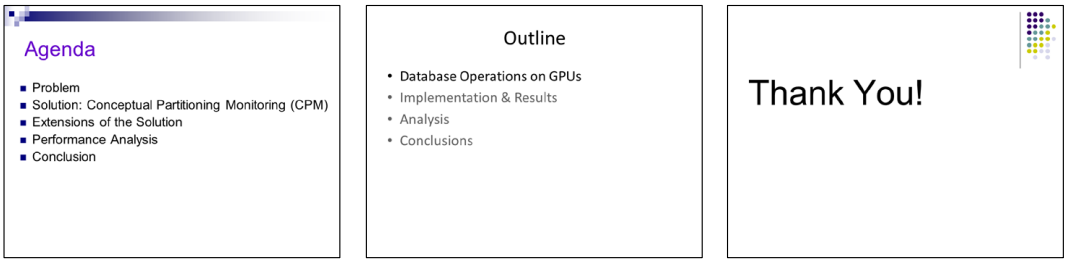


Figure 3.1: Three examples of slides from the Outline category, itself a subset of the *nil* category.

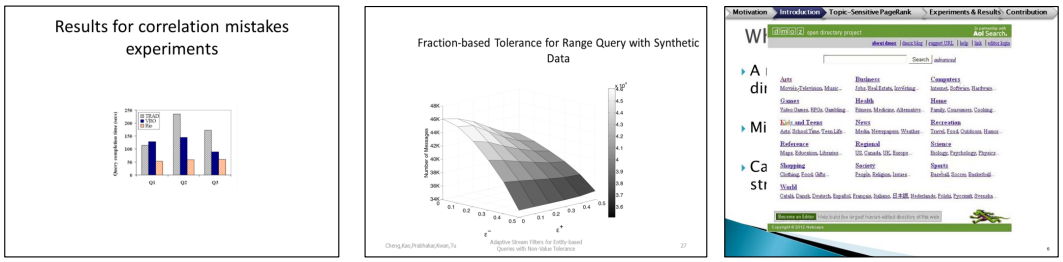


Figure 3.2: Three examples of slides from the Image category. We observed that many slides in this category reporting study results.

that specific category. For example, Image slides appear in 95% of the presentations in the dataset. In contrast, Table slides were present in just 25% of presentations, and accounts in whole for only 1% of all slides in the dataset.

Table 3.2: Slide categories and their frequency, present in the dataset.

| Slide Category | Present in number of presentations (out of 20) | Number of slides (out of 751) |
|---|---|---|
| *nil* | 19 (95%) | 128 (17%) |
| Outline | 8 (40%) | 36 (4.8%) |
| Image | 19 (95%) | 90 (12%) |
| Table | 5 (25%) | 8 (1%) |
| Drawing | 12 (60%) | 65 (8.7%) |
| Text | 20 (100%) | 409 (54.5%) |

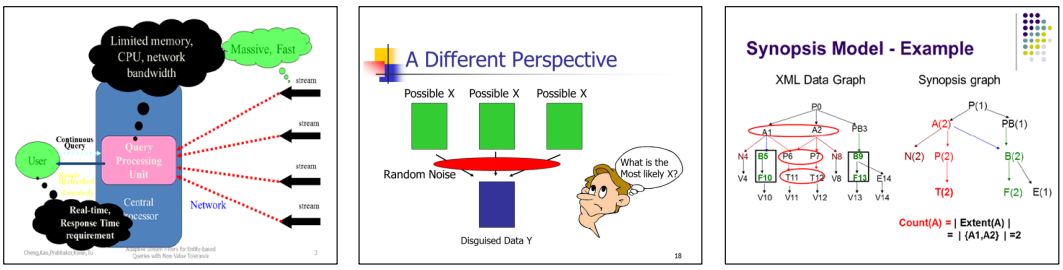


Figure 3.3: Three examples of Drawing slides.

## 3.2 Baseline Error Analysis

To understand the weakness of previous studies performance, we implemented a basic, text-only alignment system informed by the previous work. Employing standard textual similarity (cosine similarity with $tf.idf$ weighting), we aligned the sections of the document to each slide to characterize performance.

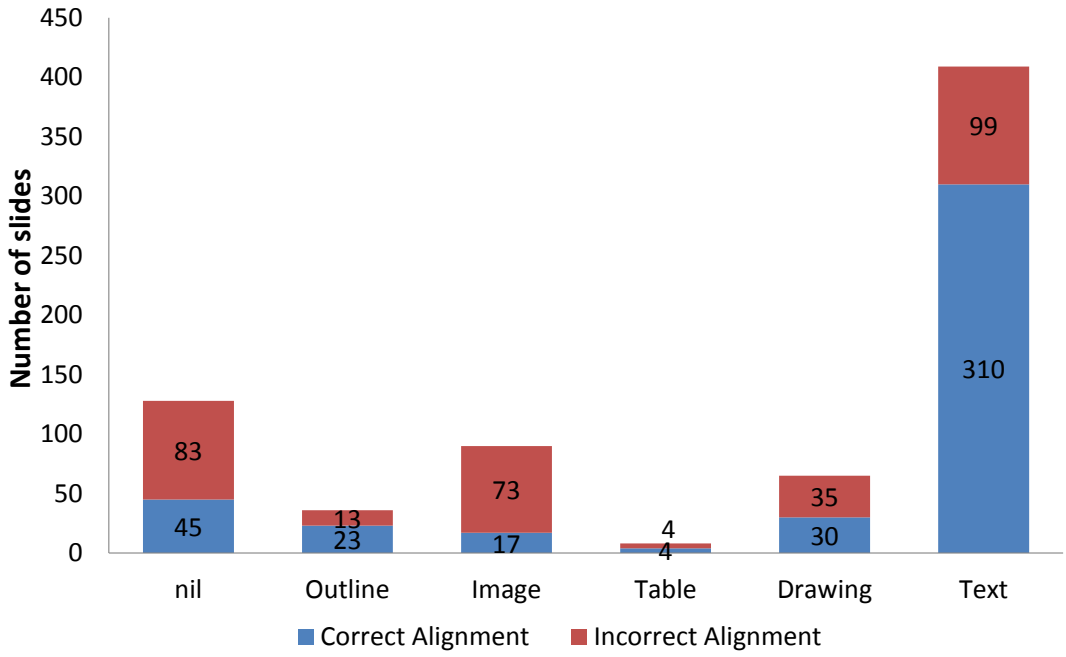


Figure 3.4: Error analysis of text-based alignment implementation on different slide categories. Text slides show relatively less error rate in compare with others.

Our findings are reported in Figure 3.4. Importantly, we find that the per-

formance of text-only alignment is not uniform; it varies per slide category. It is shown that slides from Text category are largely aligned to their correct respective sections. Earlier, we observed that both *nil* and Image categories are found in the large majority of presentations. We see from the figure that these two categories also constitute a large number of errors in the baseline. These are the error types we target to ameliorate by our multimodal technique. Identifying Image slides as well as aligning them to their related section in the paper is challenging. In addition identifying *nil* slides and ignoring them in the output alignment map is another task which has to be done to improve the alignment to an acceptable result.

# Chapter 4

# Method

To address the weaknesses of the text-only baseline, especially in aligning Image and *nil* slides, we propose a multimodal alignment methodology, which additionally leverages the visual cues and appearances of slides.

We demonstrate a general view of our system architecture on Figure 4.1. It is shown that the process starts when a new pair of paper-presentation is given to the system. Our methodology for the rest consists of three main steps, which we detail in turn:

1. Preprocessing: Every new presentation-paper is processed to extract their textual content. Text extraction from each medium has its own work flow. Subsequently, we apply part-of-speech tagging and only retain words with particular POS tags and perform other specific forms of cleanup. Section 4.1 describes this step in detail.

2. Image classification: We then classify the slide images into four pre-defined slide classes, based on our previous classification scheme. We employ machine learning to train a learner on a manually-gathered and annotated dataset of slide snapshots. Section 4.2 details the supervised training process and evaluation.

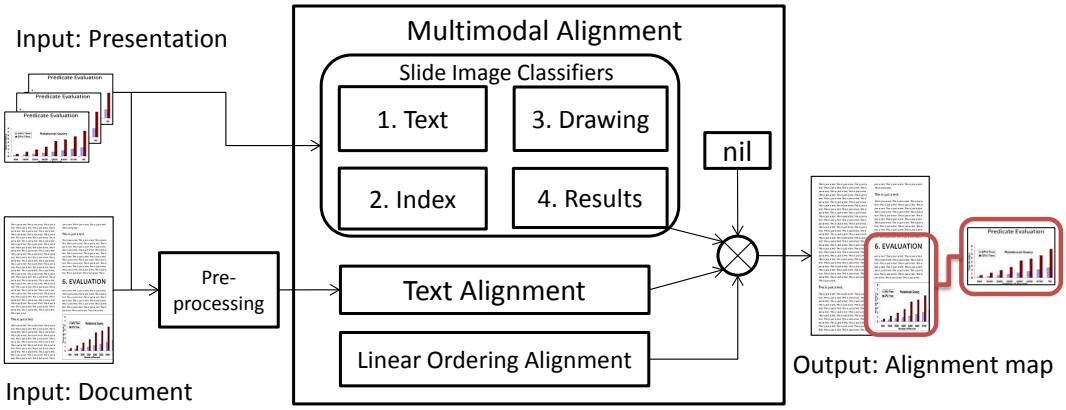


Figure 4.1: Multimodal alignment system architecture.

3. Multimodal alignment: Alignment vectors are generated for each different source of evidence: text, image and monotonic ordering preference. The image classification result plays a key role in helping to define the relative importance (probability) of each modality in the fusion process in the final alignment. This is the core alignment process and is discussed in detail in section 4.3.

## 4.1 Preprocessing

Our system presupposes the presence of text extracted from both the presentation and paper. While the documents in the dataset are born digitally, extracting their textual data is a noisy process. We spent much work in creating a pipeline to engineer relatively clean output text. To achieve this goal two steps are necessary: 1) to extract the text from input documents, and 2) to normalize and de-noise the text.

### 4.1.1 Text Extraction

#### 4.1.1.1 Paper Text Extraction

Previous works (Ephraim, 2006; Kan, 2007; Hayama, Nanba, and Kunifuji, 2005) took advantage of different PDF to XML convertors like pdf2html[1] or pdftotext. They mention the lack of accuracy and extra noise generation in this task. In (Ephraim, 2006) for example, Ephraim needs his papers' text to be extracted accurately to the paragraph level; however, he reports many failures in detecting the paragraphs – such as extractions that combine two together, or subdividing a single paragraph into two separate ones. Since we are using the same dataset, we validate his observation of much noise on the text, because of either images and tables or failure of OCR to detect the correct word. Our preprocessing pipeline however, receives academic papers with PDF format and converts them to XML format using the PDFx package[2]. PDFx is a system developed by researchers from School of Computer Science at the University of Manchester. It is specially designed to convert PDF scholarly papers to an XML format that largely preserves the paper's title and text, and importantly, recognizes sections, as well as figure and table captions. Informal comparison of the PDFx output with other text extraction/conversion systems mentioned above showed a significant improvement. For example the system identifies most figures and tables –which are an important source of noisy text extraction in previous works– and store them together with their caption. However in the cases of equations and algorithms, the extraction process still produces noise

Using PDFx, we extract the title of each section of the paper followed by the textual content of that section, and store the results as plain text files.

[1] `http://pdftohtml.sourceforge.net/`
[2] Available at `http://pdfx.cs.man.ac.uk/`.

#### 4.1.1.2 Slide Text Extraction

We extract textual information from the presentation's PowerPoint file. For each slide, we capture its title, body content, and slide number. As several options are possible, we investigated the efficacy of each alternative: 1) Converting all of the slides into .PDF, and re-using our PDF to text pipeline for paper text extraction; 2) Exporting the slide content as HTML, XML or RTF format, and then extracting the text within the exported formats; 3) Using Microsoft's Office API, which gives access to PowerPoint documents through its standard Document Object Model (as was done in (Ephraim, 2006)); 4) Using Microsoft PowerPoint's built-in Visual Basic. Our conclusion was that the final fourth methodology yielded the most accurate text extraction results.

We take each slide's title and body text, as well as the slide number, and save them in our database. As we showed before, many slides are categorized under Text slides which mainly contain text. These textual information from slides are extracted with good quality –less noise–, however for slides which contain tables, drawings and images, noise is also occasionally generated. In the case of Image slides, we obtain any text that is still available in the slide (i.e., title). In the case of Table and Drawing slides, we often obtain the text (e.g., numbers) in the many individual textboxes, which we deem mostly as noise, as they do not assist in the alignment process (e.g., Figure 3.3). Take note that text extraction from slides attempts to extract digitally stored text within each slide and it does not utilize any OCR system to extract textual information from tables, drawings and text which are stored as image.

### 4.1.2 POS Tagging, Stemming, Noise removal

Word stemming is commonly used in information retrieval systems to partially address the vocabulary mismatch problem (Metzler, Dumais, and Meek, 2007).

Beamer and Girju employ stemming in our task (Beamer and Girju, 2009), pre-stemming words before calculating text similarity. We validate their claim that it has a positive effect on the result. In our implementation we use the *Stemmer* method available in the Natural Language ToolKit (NLTK).

Stop word removal for text processing has been suggested in many studies (Huang, 2008; Metzler, Dumais, and Meek, 2007; Ephraim, 2006). Others studies (e.g., (Church, 1988; Hu and Liu, 2004; Beamer and Girju, 2009)) remove more than just stop words. They claim that part-of-speech tags have different values and effectiveness while processing text; as such, if we remove the less important tags which do not help for gaining better similarity accuracy and retain the rest which are important tags, we will get better similarity result (Beamer and Girju, 2009). These important POS tags are "Noun", "Verb", "Adjective", "Adverb" and "Conjunction". We follow these suggestions and implement such preprocessing: we tag all of the words from each paper and remove the words which are not one of the aforementioned tags. This process removes more than 1/5th of the extracted text, yet our final accuracy was improved, as discussed in more detail later. Finally, as some slides have small textboxes that contain little amounts of text that contribute to noisy alignment, we employ a simple but robust rule to remove all one- and two-character long textboxes from slides.

The result of the preprocessing is a pair of text output for the paper and presentation that has been filtered for noise, and contains stemmed words belonging to just the specific POS tags.

## 4.2 Image Classification

In the previous chapter, we conducted an error analysis on the text-only alignment baseline results that showed a high error rate on slide categories that contain visual cues (i.e. Image, Table and Drawing). Not only are a large amount of them aligned

incorrectly (Figure 3.4), but also they make up more than 20% of our dataset (Table 3.2), which attests to their importance in the alignment process.

To address this weakness, we implement an alignment system that makes use of visual information. To encode visual information, we devise a slide image classification, aimed to distinguish four easy-to-differentiate slide image classes. Note that this classification overlaps but is not identical to our earlier, baseline error-driven analysis (Chapter 3). The four classes covered by the classifier are: 1) *Text*, 2) *Outline*, 3) *Drawing* and 4) *Results*. The definition of each class are as follows:

- *Text* slides are those that are full of text. These slides do not have or have very small pictures or tables on them;
- *Outline* is the class of slides which are outline or agenda, starting of new section, ending the presentation (i.e. Q&A and Thank-you slides). This class is identical to the earlier Outline category from Section 3.1;
- *Drawing* slides are those which contains drawing shapes (i.e. texboxes and arrows). This class is also identical to the Drawing category from Section 3.1;
- *Results* slide images encompass charts, tables, and other visual objects that typically appear in the evaluation portion of a presentation. This class is a subcategory of the aforementioned Image slide category.

Our system classifies each slide into one of these classes and then based on that decides what methods to apply on the slide utilizing different modals and their weights. To be able to implement the classifier, we first train a supervised learner, detailed next.

### 4.2.1 Classifier Design

The Support Vector Machine (SVM) is a binary classifier which looks for an optimal hyperplane as a decision function. Once trained on images annotated to be in one particular class, the SVM classifier can make decisions regarding the existence of new test images in that class considering the features of the training and test images. (Chapelle, Haffner, and Vapnik, 1999) claims that for histogram-based image classification, SVMs outperforms other classification approaches since it generalizes well. We thus adopt SVM for our classification task. We manually annotated a dataset of 750 slides into the four above-mentioned classes to build a dataset for training and testing. Snapshots of every slides in each presentation was taken and stored in separate PNG files to compile the dataset .

Since SVM, by default, is a binary classifier, we produce a separate classifier for each of the four classes, fusing their judgments to arrive at the final image class. For example to build the Text slide classifier, we give all of the images to the classifier, with all of the text slides annotated as "1" (positive) and all other slides (the three other classes) as "0" (negative).

We use 10-fold cross validation to fully exploit the dataset's annotations, training four linear SVMs per fold. For each test slide, we give the image of the slide to each of the classifiers and deem the result with the highest probability as the joint classifiers' decision.

The most important issue about an image classifier to give good results is feature set. Different feature sets were discussed in section 2.3. In the following section we use pixel value, image gradient and HOG as 3 feature sets that we test their efficiency on this task.

Table 4.1: SVM slide image classification performance by feature set.

| Feature Set | Pixel Value | | | Image Gradient | | | HOG | | | HOG (preprocessed) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Slide Class | R | P | F | R | P | F | R | P | F | R | P | F |
| Text | 0.84 | 0.53 | 0.65 | 0.49 | 0.45 | 0.46 | 0.54 | 0.83 | 0.65 | 0.89 | 0.84 | 0.86 |
| Outline | 0.50 | 0.96 | 0.65 | 0.70 | 0.82 | 0.75 | 1 | 0.92 | 0.95 | 1 | 0.94 | 0.96 |
| Drawing | 0.39 | 0.91 | 0.54 | 1 | 0.82 | 0.90 | 1 | 0.82 | 0.90 | 1 | 0.82 | 0.90 |
| Result | 0.50 | 0.94 | 0.65 | 1 | 0.83 | 0.90 | 1 | 0.83 | 0.90 | 1 | 0.83 | 0.90 |
| Average | 0.55 | 0.83 | **0.62** | 0.69 | 0.75 | **0.75** | 0.88 | 0.85 | **0.85** | 0.97 | 0.85 | **0.90** |

### 4.2.2 Image Classification Results

We assessed different input feature sets for their efficacy in the image classification task. Using just the simple feature of pixel value (each pixel's value is considered as a feature; so for each slide image, there will be $960 \times 720$ individual features) is the simple baseline. Results are shown for four individual different classifiers by recall, precision and $F_1$. Last row shows the average recall, precision and $F_1$ on all classifiers. Average $F_1$ on 4 classifiers are reported as 62%.

In section 2.3 we mentioned that there has been some studies suggesting the usability of image gradients for text extraction from images and video frames (Ye et al., 2005; Dutta et al., 2009) Thus, for a second feature set, we use image gradients. The number of features in this case is same as before (equal to number of pixels). We applied a $3 \times 3$ Sobel mask on the image to obtain the gradient. This actually extracts the edge of the images. Using the image gradient value for each pixel, instead of the actual pixel value, causes better classification results as can be seen in Table 4.1. The average $F_1$ increases 13%, and recall for two classes obtain the highest results, but in Text class results are not good. One possible reason is that text contents, when taken image of, are usually narrow. Thus after detecting the edges, just bold and thicker fonts will remain in the image.

Given the promising results obtained by image gradients, we further explore more discriminative gradient-based features exploiting Histogram of Oriented Gra-

dients (HOG) (Dalal and Triggs, 2005). HOG improves the gradient features by a two phase approach – first, utilizing gradient orientations voted by gradients magnitudes (known also as pooling), and in a second phase, by employing spatial blocks normalization. In this work, we perform the first phase, gradient voting, using small spatial patches with patch size of 9. We also realize that block normalization slightly improves the classification accuracy but with higher computation complexity. Therefore, we leave the second phase for future work when more comprehensive dataset is available.

We computed HOGs for each slide image and used them to train a linear SVM, achieving an improvement of around 10% on $F_1$. For this task no preprocessing was done on images and raw images were given to the learner. Though the results can still increase with some simple preprocessing techniques. We have applied two: 1) power normalization, and 2) boxfilter blurring. Power normalization enforces each image to have a mean of 0 and standard deviation of 1. Boxfiltering reduces the side effects caused by noise inputs and high contrast between slide's contents and its background. Take note that the parameters in HOG implementation were optimally tuned for the best result (patch size = 9 and bin size = [32 32]). Classifier performance after images preprocessing are shown in Table 4.1. The results show that even such a simple image classifier that relies only on HOG features returned an acceptable average $F_1$ measure of 90% over our cross-validation runs. This result is enough for the image classification to be relied upon in our downstream alignment task.

## 4.3 Multimodal Alignment

Multimodal systems utilize evidence obtained from different modalities. Our method uses the result of the image classification as the key evidence that dictates how the remaining multimodal evidence is fused to form the final decision.

Our system models the two evidence sources of 1) textual similarity and 2) natural, linear ordering as generating a distribution of possible alignments for each slide. Given a slide $s$, we process both sources through modules to output a vector $v_i$ of length $|D|$ that represents the probability of aligning the slide to the particular document section $d_i$. We then fuse these vectors (marked as $\otimes$ in system architecture, Figure 4.1) into a final alignment using heuristic rules that use the image classifier's results as key evidence.

### 4.3.1 Text Alignment

We compute a cosine similarity between each slide $s$ and each section $d$, using $tf.idf$ weighting, as is recommended by the prior work (Beamer and Girju, 2009; Kan, 2007) for the component values of our text similarity alignment vector $v_{Ts}$. The output of this similarity score has no upper bound limit; we thus normalize the vector to unity to form a probability distribution. We take the maximal value within the vector as the most probable alignment point. Algorithm *TextSimilarityAlignment* shows the pseudocode for this computation. Take note that all previous studies restrict themselves only to this form of textual evidence (Hayama, Nanba, and Kunifuji, 2005; Beamer and Girju, 2009; Kan, 2007).

While our system adopts the best practice with respect to text similarity methods from previous work (Kan, 2007), we adopt Hayama *et al.*'s (Hayama, Nanba, and Kunifuji, 2005) decision to use sections as the unit of granularity for the paper, as we observe that many paragraph spans from (Kan, 2007) actually covering almost all paragraphs of a section/subsection, validating the section – rather than the paragraph – as the most nature alignment unit.

**Algorithm** *TextSimilarityAlignment*
1. $S_{1-n}$ ←text of $n$ slides from the presentation
2. $D_{1-m}$ ←text of $m$ sections from the paper
3. **for** $s \in S$
4. $V_{Ts}$ ←Text alignment vector for slide $s$, initially **nil**
5. $W_s$ ←words of slide $s$
6. **for** $d \in D$
7. **for** $w \in W_s$
8. $tempVector_w \leftarrow tf.idf(w, d, D)$
9. $V_{Ts,d} \leftarrow average(tempVector)$
10. $sum$ ←Sum of all cells in $V_{Ts,d}$
11. $V_{Ts,d} \leftarrow V_{Ts,d}/sum$
12. return $V_{Ts}$

Figure 4.2: $tf.idf$ cosine text similarity computation for a slide set $S$ and a document $D$. The average $tf.idf$ score of slide $s$ with first section of the paper, is stored in the first cell of vector $v_{Ts}$. Similarly score of this slide with next section is stored in next cell. So vector $v_{Ts}$ has the length of $|D|$ and shows the similarity of slide $s$ to different sections of the paper.

### 4.3.2 Linear Ordering Alignment

Kan claims that slides follow a monotonic alignment progression with respect to the paper flow (Kan, 2007). He implemented several alignment methods including: maximum similarity, edit distance and a local jump model. Maximum similarity is a greedy model that does not model any monotonic preference and the target paragraph can be selected from anywhere in the paper. In edit distance, a dynamic programming approach is used to maintain an optimal path that penalizes deviations from the monotonic path (Kan, 2007). The weakness is that it cannot align slides to previously skipped sections. Kan solves this weakness by implementing local jump model which relaxes the restriction and allows alignment with recently passed sections. However, he stated that it only adds overhead to the search space and does not improve results.

Studying our corpus, most pairs show that the ordering between slides and sections are monotonic which validates Kan's claims (Kan, 2007). The gold-standard alignments of slides to paper sections in our collection is shown in Figure 4.3. For the purpose of this analysis, the number of slides and sections of all presentations are scaled to 26 and 37, respectively (the average number of slides and sections on the dataset). Then according to the truth-ground annotation of the dataset, we count the number of presentations that have the same alignment for each cell. Darker cell values indicates a larger number of alignments that fall in the cell. As can be seen from the figure, most presentations have a monotonic alignment tendency. I.e., slides at the beginning of a presentation are most probably aligned to early paper sections.

To model the preference for monotonic alignment, we encode an alignment probability vector $v_{Os}$. This vector gives the linear mapping of $M$ sections to $N$ slides the highest probability. The linear mapping for slide $s$ is calculated by $\lfloor \frac{s}{N/M} \rfloor$. We assign smaller alignment probabilities to close neighbors of the exact linear

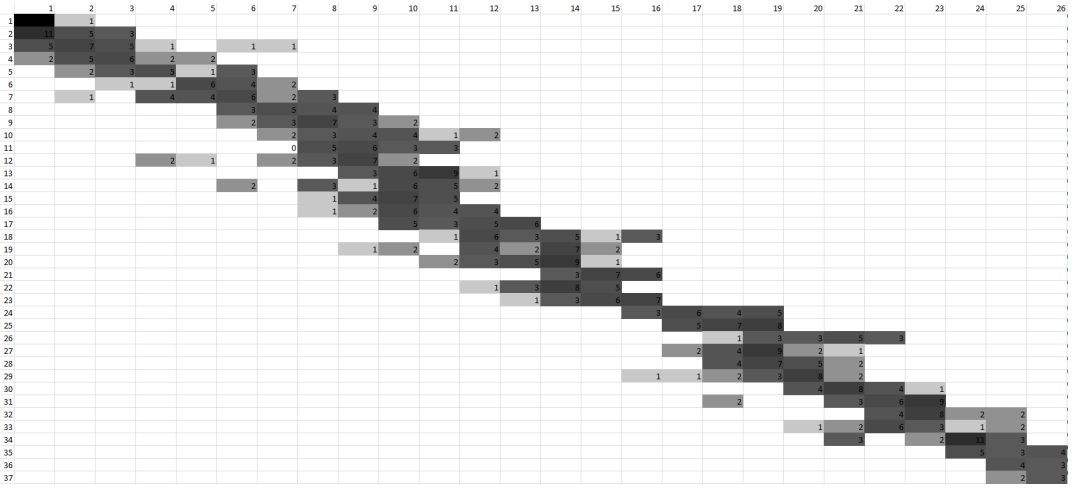

Figure 4.3: Visualization of alignment map for all presentations. Rows represent slides and columns represent sections. Sections and slides of each pair are scaled to fit in the current number of rows and columns. Darkness is in accordance with the number of presentations which fit in the same alignment.

mapping to smooth out the alignment point, as individual slide-paper alignment pairs do deviate from the norm and may require local jumps off of the true diagonal, as in Figure 4.3. Document sections distal to the linear alignment point are assigned zero probability in this vector. In specific, we model the probabilities heuristically. In the linear alignment vector $v_{Os}$, the computed linear alignment point's cell is given a value of 0.4, its neighbors are given 0.2, and its neighbors' neighbors are given 0.1. Figure 4.4 shows an example of a paper with 9 sections and for which the linear ordering assigns the most likely alignment point to be the fifth section (section 3.1). If the centric point is calculated to be the first or the last one, the sum of the probability of the two cells that fall out of the range (0.2+0.1) is divided into 3 remaining cells topping up their probabilities to 0.5, 0.3 and 0.2 respectively.

### 4.3.3 Slide Image Classification-based Fusion

To fuse the results, we have two input vectors for an input slide $s$: $v_{Ts}$ and $v_{Os}$. We define three weights – $w_{Ts}$, $w_{Os}$ and $w_{nil}$ – which are the weights assigned to the


Figure 4.4: An example of a linear alignment vector in a 9-section paper, where the most probable cell for alignment is the 5th cell (section 3.1). Values in each cell indicates the probability assigned to that cell(section). The underside row shows the section numbers extracted from section title.

importance of the textual similarity, linear ordering and *nil* alignment, respectively. These three weights are initially equal and sum to unity ($w_{Ts} = w_{Os} = w_{nil} = 1/3$). How we fuse the results depends categorically on the image classification results for the slide. We review the fusion methodology for each of 4 cases of the image classifiers' possible outputs:

- *Text* class. When $s$ is "Text", it is deemed to mainly consist of text. Text similarity measures are most accurate for these cases (when there is sufficient amount of text is available). We thus want to increase the weight for text alignment vector ($w_T$), scaling for the amount of text present on the slide. For each presentation $S$, we count the number of words for each slide $s$ and take the maximal count as a full text slide. We thus express $s$'s count of words as percentage of the maximum and assign $w_{Ts}$ as:

$$\Delta w_{Ts} = w_{Ts} \times (1 + \frac{wordCount(s)}{max\{wordCount(s) : s \in S\}}) \tag{4.1}$$

  As examples, let $S$ have two slides $i$ and $j$ that are classified as "Text" with 75 and 50 words, respectively. If $i$ has the maximum number of words (i.e., 75 words) for all slides in $S$, $\Delta w_{Ti} = w_{Ti} \times (1 + (75/75)) = 2w_{Ti}$ and $\Delta w_{Tj} = w_{Tj} \times (1 + (50/75)) = 1.66 w_{Tj}$. Take note that while $w_T$ is scaled up, we leave $w_O$ is fixed, resulting in $w_{nil}$ shrinking.

- *Outline* slides are potentially *nil* slides, as the image classifier's "Outline" label signifies an outline/index/agenda/thank-you slide. Our system scales

down both vectors $w_T$ and $w_O$ by 0.66 to discourage alignment. The resulting values are thus $w_T = w_O = 2/9$ and $w_{nil} = 5/9$. This will increase the probability of slide $s$ not to be aligned to any section.

- *Drawing.* For "Drawing" slides, we cannot draw any conclusion from the slide's visual appearance – we could not find any bias towards a favored modality for alignment–. We set the weights in this case uniformly: $w_T = w_O = w_{nil} = 1/3$. In this case, the system trusts other modalities for their alignment judgment, instead of evidence from the slide image.

- *Result* class. "Result" slides are mainly charts and tables, and usually exhibit a small amount of words, making them difficult to align via textual means, as demonstrated in Figure 3.4. We observe that almost all slides with charts, diagrams and tables are related to "Experiments and Results" sections of a paper, and unconditionally align these slides to the according section (if one exists). To decide which paper section represents the results, we use a simple regular expression based approach. We seek a section header that matches any of the following lexical patterns: {"Result", "Experiment", "Evaluation", "Discussion"}. In our limited experiments, we can locate the correct result section in about 95% of papers.

**Nil Classifier**

Kan (Kan, 2007) suggested a *nil* classifier. He structured the challenge as a supervised machine learning problem. For the supervision, he used the cosine text similarity score (as is used in our text similarity) and the number of words present on the slide. For our *nil* classification, we also use these two features, but in an unsupervised way:

$$p(nil) = 1 - \frac{max\{similarityScore(i,d) : d \in D\}}{max\{similarityScore(s,d) : s \in S, d \in D\}} \times \frac{wordCount(i)}{max\{wordCount(s) : s \in S\}} \tag{4.2}$$

where $D$ is the collection of all sections in the paper and $S$ is the collection of all slides in the presentation.

Giving $w_{nil}$ that was computed according to the slide image classification result and $P(nil)$ described above, we define $w_{nil} \times P(nil)$ as nil factor. We set the nil factor threshold to 0.4 since we observed most *nil* slides have nil factor higher than 0.5. We then apply the nil factor threshold on slide. If the slide $s$ does not fall in *nil* category –nil factor lower than 0.4–, we fuse the weighted text and order alignment vectors as a final alignment vector (FAV):

$$FAV = w_{Ts}(v_{Ts}) + w_{Os}(v_{Os}) \tag{4.3}$$

The maximal value of FAV cell gives the final, selected target section that the slide is computed to align to.

# Chapter 5

# Evaluation

We first describe the evaluation methodology and then report text-only baseline results. We perform feature efficacy testing, by incrementally adding one feature at a time to record the change in performance. We end with a discussion on the alignment performance per slide category.

## 5.1 Experiments and Results

Our experiments reuse Ephraim's dataset (Ephraim, 2006), which we modified to suit our needs. We added annotations to include the alignment key (ground-truth) between all slides and their respective sections. However for the first experiment, we use the same annotation (slide-paragraphs) that was done by Ephraim in the dataset – i.e., paragraph-to-slide alignment, instead of section-to-slide alignment–. For this initial experiment, we only used textual data to compute the probability vector. (Ephraim, 2006) and (Kan, 2007) performed the same experiment on the dataset; their best results are reported alongside ours in Table 5.1. We performed preprocessing as described in Section 4.1.2: stemming, POS tagging and filtering unwanted POS tagged words.

With our baseline implementation, we achieve an accuracy of 52.1%, outperforming Kan (Kan, 2007) experiment using the same. Introspection the results, we believe the reasons are: 1) our preprocessing pipeline uses more accurate text extraction tools for both slides and papers which results in less noisy data; and 2) Kan employed a different evaluation method of Weighted Jaccard accuracy, which penalizes result when it has less overlap with the correct answer. In our proposed system however, a slide is correctly aligned if the first suggested paragraph is correct. (Ephraim, 2006) reports 62% for his best result which was achieved by Lucene similarity measure. All of the results are shown in Table 5.1.

(Hayama, Nanba, and Kunifuji, 2005) suggests slides to be aligned to sections instead of paragraphs. It is expected to show better results since sections are more coarse-grained. To confirm that, as our second experiment, we performed another bi-modal alignment which aligns slides to sections. For the evaluation, we counted the number of slides which were correctly aligned to their respective sections. Table 5.1 shows that coarser-level granularity yields a perceived improvement of nearly 8.5%.

Table 5.1: Alignment accuracy results for different experiments. Note that several of these results are not strictly comparable.

| Method | Accuracy |
|---|---|
| Kan (weighted Jaccard)(Kan, 2007) | 41.2% |
| Beamer (original results)(Beamer and Girju, 2009) | 50.0% |
| Experiment 1: Paragraph-to-slide | 52.1% |
| Experiment 2: Section-to-slide | 60.7% |
| Ephraim(Ephraim, 2006) | 62.0% |
| Experiment 3: Exp. 2 + Order alignment | 66.8% |
| Beamer (manual *nil* removal)(Beamer and Girju, 2009) | 75% |
| Experiment 4: Exp. 3 + Image Classification | 77.3% |

Experiments 1 and 2 are considered baseline experiments. In our third experiment, we complement the text similarity baseline with the influence of ordering alignment. In this multimodal alignment, we gave static uniform weights to both

probability vectors. The result of this experiment showed an improvement of 6% which was obtained by taking account the monotonic order of slides and sections.

We perform Experiment 4 to analyze the effect of the image classification system – which includes the unsupervised *nil* classifier – on the previous results. In this experiment, we use the full functionality of our multimodal alignment, improving the results by an absolute 10.5%. As demonstrated in Table 5.1, we achieved more than 77% accuracy, a large improvement over the first (52%) and second baselines (60.7%). While our results are not directly comparable, our results indicate a higher accuracy when compared with Beamer *et al.* (Beamer and Girju, 2009), although they removed *nil* slides manually, and used a different dataset.

Take note that in Experiment 4, each slide is given to the image classifier and according to the image class that is assigned to that slide, further steps of multimodal alignment was taken place. For the image classification result to be fair and valid, we took two pairs of presentations and papers as one of ten folds for cross validation. We trained the image classifier with slide images from the remaining 18 pairs. We then use slide images of these 2 presentations as test set and classify them. After that their slides are classified, system applied the other necessary processing to obtain the target section for each slide. Checking the returned section with the annotated alignment key, we consider the alignment as correct or incorrect and calculate the percentage of corrects alignment for that 2 pairs. This procedure is done 10 times and each times with 2 new pairs until all pairs have been considered as test set once. Taking the average of the correct alignment for each pair, we calculate the final accuracy of Experiment 4 (our best result).

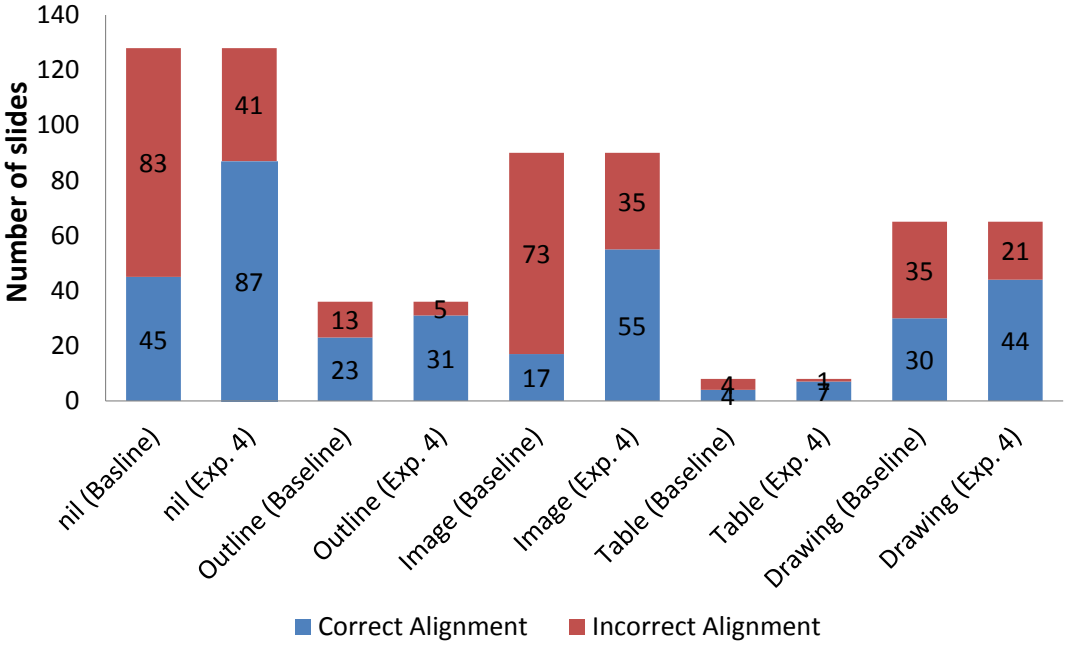


Figure 5.1: Error rates of the baseline (l) and proposed multimodal alignment (r), broken down by slide category.

## 5.2 Discussion

We break down the performance gains by our system by image class, to dissect and explain the changes in alignment performance and to identify opportunities for future development. We plot Figure 5.1, which places performance of the baseline and our best system side-by-side (cf. Figure 3.4). For each category, the left bar in the pair shows the number of slides which were aligned (in)correctly by the baseline, whereas the right bar shows the same information for the full multimodal system (as given by Experiment 4).

It can be seen in the figure that error rate in all categories have decreased significantly. However, there are still incorrect alignment in the results. We describe these in detail:

- 42 of the incorrectly aligned *nil* slides are now correctly deemed as *nil* by our proposed system. Our *nil* classifier improved accuracy alone by over 5.5%,

confirming our initial assumption about effects of *nil* classification from previous works. Kan (Kan, 2007) reports 3%, Hayama (Hayama, Nanba, and Kunifuji, 2005) reports 3.4%, Ephraim (Ephraim, 2006) reports up to 11% and Beamer (Beamer and Girju, 2009) reports up to 25% of improvement can happen by implementing a *nil* classification system. Our study pegs this number at 5.6% as can seen in the leftmost pair of bars in Figure 5.1. Note that according to Table 3.2, around 17% of slides are *nil* and our system identifies more than 11.5%, which we feel is acceptable. The remaining 5.5% incorrectly aligned slides are mainly ones with large amount of text, and common words with the sections, but not related or extracted from them. In these cases, our system gives a high weight to text similarity, which discourages *nil* alignment.

- The next two bars report "Outline" error rates, which are a subset of the first columns. Thus, the improvements here are already counted in *nil* category before. The figure shows that just 5 slides are incorrectly aligned in this subcategory. Our investigation shows that although these slides are correctly classified as Outline, their nil factor fall below the threshold. The reason can be both number of words ratio or text similarity score ratio which penalizes *nil* classification.

- The next two bars are for "Image" category. Here, we see large improvements. The number of incorrectly aligned slides (73 in the baseline), is decreased by almost half (35 in Experiment 4). As observed and reported in earlier sections, many Image slides actually report experimental results. Our image classifier tends to identify those specifically include charts and tables and aligns them to their respective section. The 38 image slides which are correctly aligned in our system (55 in total) as well as 3 correctly classified Table slides, shows the effectiveness of our method on Image slides. However, there are still 35 Image slides which remain incorrectly aligned. Our microscopic analysis reveals that

more than half are slides which contain images of the text from the paper. Figure 5.2 (a) is an example, where the slide has been correctly classified as a Text slide, but there is no any digitally stored text on that to be extracted. Being classified as Text slide is pushing the system to trust the text similarity alignment, however due to lack of textual data, the text alignment produces incorrect results. An additional type of error is when a slide which contains chart or table, does not report results or experiments. We observe that they may report analysis done earlier in the paper. Figure 5.2 (b) is an example of a slide which according to its visual content is aligned to "Results" section, incorrectly.

- The number of incorrect alignment in the "Drawing" category has also been decreased. In the case of Drawing slides, our system gives uniform weights to different alignment probabilities ($w_T$, $w_O$ and $w_{nil}$). In addition in the baseline (left bar) we also used the same text data of the slides, therefore this can be inferred that the improvement in this category is mainly because of the suggested ordering alignment.

- 99 Text slides were aligned incorrectly according to baseline analysis (Figure 3.4). After the multimodal alignment is done in experiment 4, the number of incorrect alignment decreases to 70. Although our text similarity measure has not been changed, we can see a significant 4% of improvement in the final results caused by Text slide. Monotonic alignment can justify this improvement. Take note that the Text slide results were removed from Figure 5.1 due to large difference on scaling with the other categories.

In many of the previous works mentioned before, it is concluded that *nil* classification is necessary, however none of them implement this functionality, except for (Kan, 2007). In addition, to our best of knowledge, in almost none of the

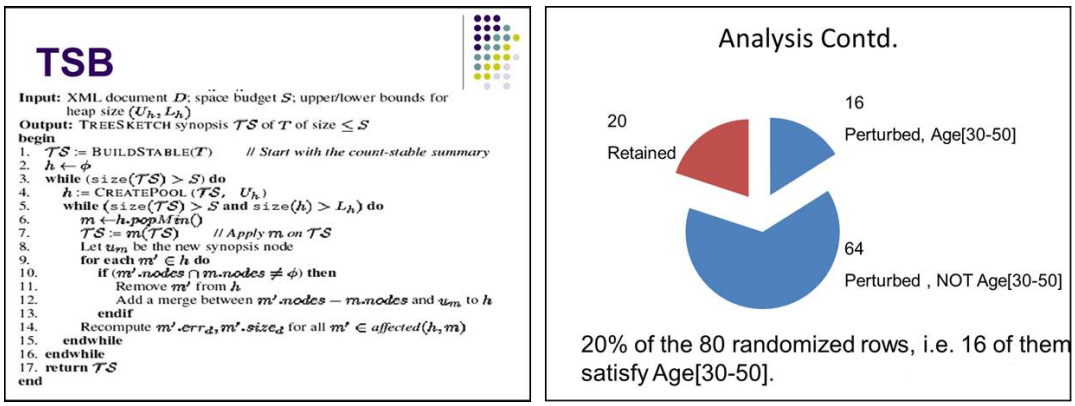


Figure 5.2: a) Left picture is an example slide containing an image of the text from the paper. These slides are a source of error as the image classifier correctly puts them in the Text class. But the content is an image of text, instead of digitally stored text. Therefore our text extraction process locates little or no text for extraction, and thus are aligned incorrectly. b) Right picture is an example slide containing a pie chart. The image classifier decides that this slide belongs to "Result" category and therefore system aligns it to experimental sections of the paper. However it was appeared in the beginning of the presentation reporting a preliminary analysis.

previous similar tasks, the appearance and visual features of the slides were taken into account for deciding the related section in the paper. The results which were shown on Table 5.1 and Figure 5.1 prove our claims in the analysis section that the most errors are from slides with few words. We showed that by utilizing slide images the prediction of target related section improves significantly.

# Chapter 6

# Conclusion

We summarize our study, reviewing what we have done and observed, and suggest future work.

We first conducted an analysis on an existing dataset of presentations, observing that more than 40% of slides contain elements other than text. We categorized such non-text-centric slides into different six types, presenting statistics for each category. To observe how a baseline fares on these categories, we implemented a baseline that generates alignments purely based on textual similarity. The result was interesting: most errors (incorrect alignments) were from slides containing images, tables, drawing, or slides which should not be aligned (*nil*). Such non-text errors attribute to more than 26% of incorrect alignments. This is in contrast to text-centric slides which were responsible for a significantly lower percentage (13%) of incorrect alignment. This high rate of errors in non-text slides motivate us to design a multimodal alignment system which exploits appearance of the slides to complement the textual alignment.

To implement such a multimodal alignment system, we first needed to classify slide types. We designed and implemented a supervised image classifier, which uses a linear SVM to classify each slide according to its appearance. To support the

supervised learning methodology, we annotated a dataset consisting of 750 slide images. Our experimentation with different feature sets showed that histogram of oriented gradients (HOG) performed well in distinguishing slide types. The classifier distinguishes four types of slides: 1) Pure text slides, 3) Outline slides (e.g., "agenda", "thank-you" slides), 3) Drawing slides (with shapes, arrows, and textboxes), and 4) Result slides (often containing tables and charts). The highest $F_1$ measure we obtained for this image classification task was 90%.

Our final system uses the slide image classifier as a key component in its alignment. Our multimodal system takes advantages of the image categories assigned to each slide to properly weight image, text and ordering evidence in alignment. Our probabilistic system assigns a higher probability to slides when they can be monotonically aligned to their respective sections; however, other factors like text similarity can strongly influence the alignment results depending on the slide category.

The resulting multimodal system improves overall performance substantially; our system achieves more than 77% alignment accuracy, which outperforms all other previous works. Analyzing of our system's output, we find that our methodology particularly helps to identify *nil* slides. We conclude that our study has shown that visual information constitutes important evidence for document-presentation alignment which is complementary to textual similarity.

Although our work significantly reduces alignment error, there is still room for improvement. Our analysis shows that 9% of errors are unrelated to non-text slides. The alignment and similarity computation for text-centric slides need to be improved. (Hayama, Nanba, and Kunifuji, 2005) suggests to use formatting of slides for better results; similarly, (Beamer and Girju, 2009) differentiates items with bullets from other text in slides. We suggest using different weights for title and body text in slides and paper section would be useful.

To further enhance the suggested alignment model, the presence of text in slides can be more holistically leveraged as features in the multimodal classifier in the future. In the present system, we currently only use the textual data in slides in computing the textual similarity component; however, considering the text during image classification may also be helpful. For example, Outline and Result slides often contain a controlled vocabulary, whose presence could be taken as further evidence for classification (e.g., "Outline","Agenda","Overview","Index").

In a separate line of work, it is clear that more supervised alignment data would be valuable. Locating, downloading and annotating pairs of presentation and papers could improve the holistic performance of the system. In a related but separate angle, the coverage of existing system can also be improved to support additional file formats aside from PDF and PPT. In addition, an end-to-end evaluation and subsequent field study that investigates and tests the possible usage scenarios of the user interface for browsing and searching the alignments would be useful.

# References

Albiol, Alberto, David Monzo, Antoine Martin, Jorge Sastre, and Antonio Albiol. 2008. Face recognition using hog–ebgm. *Pattern Recognition Letters*, 29(10):1537–1543.

Beamer, B. and R. Girju. 2009. Investigating automatic alignment methods for slide generation from academic papers. In *Proceedings of CoNLL*, page 111.

Chapelle, Olivier, Patrick Haffner, and Vladimir N Vapnik. 1999. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064.

Church, Kenneth Ward. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143. Association for Computational Linguistics.

Dalal, N. and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *Proceedings of CVPR*, volume 1, pages 886–893. IEEE.

Dutta, A, U Pal, P Shivakumara, A Ganguli, A Bandyopadhya, and C. L Tan. 2009. Gradient based approach for text detection in video frames.

Ephraim, Ezekiel Eugene. 2006. Presentation to document alignment. Undergraduate thesis, National University of Singapore.

Fei, Wang. 2006. Synthetic image categorization. *Honours Year Project Report*.

Gale, William A and Kenneth W Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 177–184. Association for Computational Linguistics.

Gokul Prasad, K, Harish Mathivanan, TV Geetha, and M Jayaprakasam. 2009. Document summarization and information extraction for generation of pre-

sentation slides. In *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*, pages 126–128. IEEE.

Hasegawa, Shinobu, Akihide Tanida, and Akihiro Kashihara. 2011. Recommendation and diagnosis services with structure analysis of presentation documents. *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 484–494.

Hayama, T. and S. Kunifuji. 2011. Relevant piece of information extraction from presentation slide page for slide information retrieval system. *Knowledge, Information, and Creativity Support Systems*, pages 22–31.

Hayama, T., H. Nanba, and S. Kunifuji. 2005. Alignment between a technical paper and presentation sheets using a hidden markov model. In *Proceeding of Active Media Technology*, pages 102–106. IEEE.

Hayama, T., H. Nanba, and S. Kunifuji. 2008. Structure extraction from presentation slide information. *Proceedings of PRICAI: Trends in Artificial Intelligence*, pages 678–687.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Huang, Anna. 2008. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56.

Huang, Weihua, Chew Tan, and Wee Leow. 2004. Model-based chart image recognition. *Graphics Recognition. Recent Advances and Perspectives*, pages 87–99.

Jing, H. 2002. Using hidden markov modeling to decompose human-written summaries. *Computational linguistics*, 28(4):527–543.

Jinha, Arif E. 2010. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263.

Kan, M.-Y. 2007. Slideseer: A digital library of aligned document and presentation pairs. In *Proceedings of JCDL*, pages 81–90. ACM.

Lienhart, Rainer and Alexander Hartmann. 2002. Classifying images on the web automatically. *Journal of Electronic Imaging*, 11(4):445–454.

Liew, G.M. and M.Y. Kan. 2008. Slide image retrieval: a preliminary study. In *Proceedings of JCDL*, pages 359–362. ACM.

Lowe, David G. 1999. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee.

Lu, D and Q Weng. 2007. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870.

Metzler, D., S. Dumais, and C. Meek. 2007. Similarity measures for short segments of text. *Advances in Information Retrieval*, pages 16–27.

Salton, Gerard. 1984. *Introduction to modern information retrieval*, volume Chapter 4. McGraw-Hill, Inc., second edition.

Shibata, T. and S. Kurohashi. 2005. Automatic slide generation based on discourse structure analysis. *Proceedings of IJCNLP*, pages 754–766.

Sravanthi, M., C.R. Chowdary, and P.S. Kumar. 2009. Slidesgen: Automatic generation of presentation slides for a technical paper using summarization. In *Proceeding of FLAIRS*.

Swain, Michael J, Charles Frankel, and Vassilis Athitsos. 1996. Webseer: An image search engine for the world wide web. *Computer Science Department, University of Chicago TR-96-14, available from*.

van der Plas, Lonneke and Jörg Tiedemann. 2008. Using lexico-semantic information for query expansion in passage retrieval for question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 50–57. Association for Computational Linguistics.

Voorhees, Ellen M. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc.

Wang, Fei and Min-Yen Kan. 2006. Npic: Hierarchical synthetic image classification using image search and generic features. *Image and Video Retrieval*, pages 473–482.

Wang, Y. and K. Sumiya. 2012. Skeleton generation for presentation slides based on expression styles. *Intelligent Interactive Multimedia: Systems and Services*, pages 551–560.

Wu, Dekai. 1994. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 80–87. Association for Computational Linguistics.

Ye, Qixiang, Qingming Huang, Wen Gao, and Debin Zhao. 2005. Fast and robust text detection in images and video frames. *Image and Vision Computing*, 23(6):565–576.

Yih, W.T. and C. Meek. 2007. Improving similarity measures for short segments of text. In *Proceedings of Artificial Intelligence*, volume 22, page 1489. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Zhang, Jing and Rangachar Kasturi. 2010. Text detection using edge gradient and

graph spectrum. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3979–3982. IEEE.

Zhang, Wei, Gregory Zelinsky, and Dimitris Samaras. 2007. Real-time accurate object detection using multiple resolutions. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.