PodVinder

Spoken Document Retrieval for Dutch Pod- and Vodcasts

University: *University of Twente (UT)*
Faculty: *Electrical Engineering, Mathematics and Computer Science (EEMCS)*
Department: *Human Media Interaction (HMI)*
Author: *van Gils, F.M.D.M. (frank@vangils.org)*
Supervisors: *dr. Ordelman, R.J.F.*
ir. Huijbregts, M.A.H.
dr. Larson, M. (University of Amsterdam)

Version: *29 January 2008*

Foreword

"Uitdaging"[1] was the word which jumped out at me from the description of this project when I read it back in June 2006. Along with the promise of building a completely new and practical system from top to bottom I was convinced it was a perfect project for me. After some preliminary research on the subject for my Capita Selecta I decided to continue the research for my final Master project.

The first decision that had to be made was whether I would focus on a particular part of the process or would try to tackle the complete system. Due to my ambition to create a fully functioning system I chose to go for the latter. While this made it harder in terms of research it was very satisfying to see a fully operation system in the end.

In the early stages of the project, I mainly focussed on the practical task of building PodVinder. While working with dynamic content from the Internet can be very cumbersome for scientific research because of its unpredictability I have to say it also makes it a very fascinating subject for research not knowing what to expect. Once most of the practical work was done the next step was to incorporate research into the project to be able to view the work in an appropriate context. With a complete system available it wasn't too hard to find a point of research and in my enthusiasm I again tried to cover as much ground as possible. This did not always result in a clear goal for my research which in the end led to the real "uitdaging" for me in this whole project: relate my research and communicate my findings in a clear and academic manner.

I first would like to thank Roeland. While we did not agree on everything, he made me a better academic writer and encouraged me to criticise and examine my own reasoning. I am also grateful to Martijn and Martha for taking time to read and evaluate my work. I would also especially like to thank my former housemates Vinesh, Willemijn, Manon, Sanne, Dorien and Douwe for listening and creating queries for over 10 hours of podcast material. Lastly I would like to thank my girlfriend Mairéad who proof-read my thesis more than once.

[1] Challenge in Dutch.

Contents

1 Introduction

Showing an uncle in Brazil a picture of the beautiful weather on holiday or sharing a video of the new puppy with a friend in Australia has never been easier. A mobile phone with camera and Internet access is enough: take a picture or video, upload it on a weblog, picture- or videosharing site such as Myspace, Flickr or YouTube and the whole world is informed about your latest adventures. The ease with which people create and publish information nowadays has revolutionized traditional media patterns. Up to now these patterns have been characterized by a limited number of sources (e.g., public and commercial media companies) using a limited range of media (e.g., newspapers, radio and television) dispensing information at fixed times. In contrast, people now create information on the fly using technologies like mobile phones and digital cameras and publish it directly on the Internet. People can share what they want, the way they want and users can access this information at a time and in a format of their choice. The potential of this concept is that everyone can become a publisher of information, and by using the Internet for distribution, the material becomes available around the world. Information that normally would not be published by traditional sources, such as a magnificent goal scored in a local soccer game or a 5-year old opera singer from Russia, now becomes available for anybody. Whereas the big content providers firstly made information available for users, these users are now the new big content providers. The result is an enormous source of original, specialised and exclusive content.

Podcasting and vodcasting are two new technologies that are used by people to publish information and news. While resembling traditional radio and television formats they have integrated the easy publishing and on-demand principle. The shows are published via the Internet and can be downloaded by the user at any particular time. The term podcasting is a fusion of the words ‘pod’ and ‘broadcasting’. The word ‘pod’ is explained in different ways, most sources claim that it is derived from ’iPod’, the famous mp3-player of Apple Inc., while other say that is an abbreviation of ‘Play On Demand ‘ or ‘Portable on Demand’. The term podcast is defined by the New Oxford American Dictionary as: ”A digital recording of a radio broadcast or similar program, made available on the Internet for downloading to a personal audio player”. Ben Hammersley suggested the term among others in the beginning of 2004 in an article from The Guardian that discussed ’downloadable radio’ [1]. Dannie Gregoire, founder of the popular podcast directory podcast.net, used the term later that year [2] in a forum about the development of distributing audio files. It was then picked up by Dave Slusher, Dave Winter and Adam Curry, pioneers and big promoters of podcasting. On September 28 Google listed 24 results [3] for the word podcasts. 526 hits were listed two days later and 2,750 three days after that. By October Google gave more then 100,000 hits for the first time and since then the number has grown to millions [4].

Vodcasting is a term derived from podcasting. Vodcasting is based on the same principle as podcasting, but offers video instead of audio. This thesis provides a technical definition for the terms pod- and vodcast feeds and pod- and vodcasts since other researchers have failed to do so. Before these definitions can be formulated however the terms syndication and RSS (Really Simple Syndication) need to be explained. Syndication is a form of publishing on the Internet where information from

a website is summarised in a specific format and put into one file. When new information becomes available it is directly added to this file. This way, it is possible for users to take a look at this file to see if new information has become available on the website. An example is a sport website where all the information is summarised and put into one file together with title and link to the full story. When, for example, the national rugby team wins an important game this news is added to the website and a small summary is added to the file including the title of the story and the link to the full story. A user can request the file from the website, sees a new story is added, reads the description and if interested can click the link to read the whole story. Having this file available and updating it with new information is called syndication, the most commonly used standard for this file is called RSS. Based on this explanation of syndication and RSS the following definitions for pod- and vodcasting are presented in this thesis[1]:

- **Pod/Vodcast feed (show)**: A file using a syndication format, for example RSS, offering a direct URL to the published pod/vodcast(s).
- **Pod/Vodcast (episode)**: An audio/video file syndicated on the Internet using a pod/vodcast feed. The audio or video file can be directly downloaded from a URL offered in the pod/vodcast feed.

The adoption of the podcast technology has grown considerably, which has led to vast amount of information published each day. How do users find podcasts that interest them? This is mainly by the metadata (descriptive information about the content) that is available around the podcast such as the website the podcast is published on or the information available from the feed. The problem however is that the metadata is manually created and can be very limited in its description or does not reflect the actual content of the podcast. This can make it very hard to find shows or episodes using traditional search engines (e.g., Google and Yahoo) that index the podcast based on this metadata. The growth of podcasting however triggered several initiatives to improve the accessibility of podcasts. Directories are the most common initiatives (e.g. podcastalley.com and podcast.net). In these directories podcast feeds and podcasts are accumulated and categorised per topic. This gives structure to the offered material and creates a smaller search space, which makes it easier for users to find material. Other initiatives, like Everyzing[2] and Podscope[3], make use of Automated Speech Recognition (ASR). The goal of ASR is to decode speech into text. While the technology has greatly improved over the years recognition accuracy varies depending on the domain (e.g. broadcast news, discussions programs, meetings, etc.). The accuracy on English and Spanish speech however is good enough to find podcasts based on spoken language in the podcast. So people using Everyzing and Podscope, applications that also index podcasts on transcripts (the text generated by the speech recogniser), can search for podcasts based on their content. Everyzing and Podscope however only offer support for English and Spanish podcasts with no other applications available online that support other languages. An interesting question is if the development of content based retrieval of podcasts published in other languages is also feasible in terms of available material, interest in this material and technology.

[1] Where podcasts are mentioned in this thesis, podcasts and vodcasts should be understood.
[2] Everyzing, http://www.everyzing.com
[3] Podscope, http://www.podscope.com

With a system already available for content based retrieval of Dutch news broadcasts [5] an interesting next step would be an application for content based retrieval of Dutch user-generated broadcasts in the form of podcasts. The big difference between the two is the variation in content and quality of user-generated material. This is especially important considering the accuracy of the automatically generated transcriptions by the speech recogniser. Poor audio quality or difficult domains can lead to poor accuracy of transcriptions which in turn causes poor retrievability.

In this thesis the feasibility of speech-based retrieval of Dutch podcast is explored and tested in terms of supply, demand and technology. Firstly, an outline of the whole project is given in chapter two. After the outline the thesis is broken down into the following three chapters: collection, analysis and search. Each of these chapters presents a part of the whole process of making podcasts retrievable with the use of ASR. Each chapter will discuss theory and research performed in this thesis. Recommendations for future research and conclusions will be given in the last chapter.

2 Project Outline

This chapter gives a concise introduction to Spoken Document Retrieval and the research question and hypothesis discussed in this thesis.

2.1 Spoken Document Retrieval

The goal of Spoken Document Retrieval (SDR) is to make retrieval of audio recordings possible by using information from speech contained in the audio. This is done by a combination of automatic speech recognition and information retrieval techniques. First Automatic Speech Recognition (ASR) is used to generate a time-marked textual representation (transcript) of the speech inside the audio. Then the transcript is indexed and can be searched using an Information Retrieval engine. In traditional Information Retrieval the information need of a user, typically expressed in a 'query' or 'topic', is used to search the index resulting in a ranked list of relevant documents.

SDR applications make it possible to access audio and video archives (e.g., radio and television broadcasts, meetings, lectures) without the need for human-generated transcripts. This is particularly interesting in view of the growth of user-generated audio and video material on the Internet. Especially since the generated material, although mostly created by non-professionals, is a source of original, specialised and exclusive content.

2.2 Research Questions

This thesis focuses on the feasibility of an SDR system for Dutch podcasts, the possibility to use a Dutch speech recogniser in combination with a text retrieval system to create a content-based retrieval system for Dutch podcasts. The thesis is divided into three parts: *collection*, *analysis* and *search.*

Collection is the first part and focuses on the collection of Dutch podcasts. The supply, demand of podcasts is investigated to see if there is enough material and enough demand to consider the development of a Dutch SDR system. In addition, a look is taken at characteristics of the Dutch supply. What kind of material should a system be able to support?

The second part, *analysis*, focuses on the automatic generation of metadata for podcasts. What information is available and can extra information be extracted? Also the accuracy of Dutch ASR on podcast is checked to see whether the performance is good enough for retrieval purposes.

Search discusses the retrieval process of podcasts from the index. An experiment is performed to determine whether the inclusion of automatic generated metadata improves the retrievability of podcasts and if podcasts are retrievable on information extracted from the speech inside the podcast.

To summarise, following research questions are formulated:

- Is the supply and demand of Dutch podcasts sufficient to consider a SDR system?
- Is the performance of Dutch Automatic Speech Recognition on podcasts enough for retrieval purposes?
- What is the retrievability of podcasts only using user-generated metadata?
- Does indexing podcast with automatic generated metadata (by ASR and other tools) improve the retrievability of podcasts?

2.3 Hypothesis

The retrieval of Dutch audio based on information extracted from speech is possible with a system already available for Dutch professional broadcast news. The variation in content and quality of user-generated podcasts, however, is wider than that of professional news broadcast. This influences the accuracy of the generated transcripts by the ASR. The user-generated material however is supported by, although sometimes limited and incorrect, metadata. Considering this information and the main question, whether the development of an SDR system for Dutch podcasts is feasible, the following hypothesis was formulated:

Dutch podcasts can be retrieved based on a combination of information extracted from speech inside the podcast and user-generated metadata. Current and future supply of and demand for Dutch podcasts validates this approach.

2.4 Prototype

During the research a SDR prototype, dubbed PodVinder, was built. The goal of the prototype was to automatically make (newly published) material searchable. The prototype was also used to answer some of the research questions and can serve as a foundation for further research. Due to limited time and resources the first version only supports the mp3-format since it is the most common format for podcasts. To ensure that future development of the system is possible without rewriting big parts of the implementation it is flexible in terms of migration to other platforms and adding/updating features.

Technical documentation of the prototype is divided into four parts: collection, analysis, search and presentation. These parts can be respectively found in Appendix D, E, F and G.

3 Collection

In this chapter the feasibility of a SDR system for Dutch podcasts is discussed in terms of supply and demand. It is researched whether the volume, level of interest and number of downloads for podcasts justifies the development of an automatic system for analysing, organizing and searching these podcasts. The available Dutch podcasts are also checked for characteristics such as favourite format and bitrate to determine what kind of material a system should be able to handle. In order to put Dutch podcasting into perspective and see whether it follows a global trend, the international supply, demand and future of podcasting is also discussed.

3.1 Hypothesis

Based on the popularity of the creation and usage of user-generated content and with podcasting being one of the newest technologies to publish information via the Internet it can be assumed more and more material becomes available via this medium. This would imply that the podosphere, the collection of all podcasts available for download, is growing with new podcasts being added regularly. In addition, the growing interest in user-generated content would suggest that the overall interest thus the number of downloads of podcasts is growing as well. Combining these assumptions with the overall question if the development of automatic system would be a logical step the two following hypotheses were formulated.

- *New Dutch podcasts are regularly available expanding the podosphere in such a way human organising would take more time then automatic organising.*
- *The interest in Dutch podcasting is growing thus increasing the number of Dutch podcast downloads.*

3.2 Method

To validate the hypotheses two methods were used. First reports about the numbers of supply and demand were collected and popular publish sites such as PodcastAlley[1] (podcast directory) and Feedburner[2] (a provider of media distribution and audience engagement services for blogs and RSS feeds) were consulted via the Internet Archive: Wayback Machine[3] to retrieve figures from the last few years. Following this the first part of the prototype was built to gather information about the number of Dutch podcasts being published.

Podcastfeeds normally have a channel description including the type of language spoken in the podcast. As shown in figure 3.1 the feed carries a <language>-tag. Inside this tag a language code is placed. For the Dutch language the following codes are used: nl (Dutch), nl-nl (Netherlands-Dutch) and nl-be (Flemish). During the research feeds were also discovered carrying no official codes such as 'Dutch'. The spider developed for the prototype used a list of podcast news, directory and publishing website (see Appendix A for list) with both a Dutch and Belgium background as a start point to search for feeds containing Dutch language codes (official and non-official). Feeds were also manually added from both the Apple

[1] Podcast Alley, http://www.podcastalley.com
[2] FeedBurner, http://www.feedburner.com
[3] Internet Archive Wayback Machine: http://web.archive.org/collections/web.html

podcast directory and the podcast client PodSpider[1] that both offer to search for podcasts in a certain language. It can be concluded that shows which user more general Dutch and Belgium podcast sites to increase exposure were all collected. It is impossible to prove, however, what proportion of the Dutch material in the podosphere was discovered. It is possible Dutch podcasts not seeking promotion via these sites or podcasts by other Dutch speaking people (e.g. Surinamese) were not collected.

Figure 3.1: Example of Dutch Podcastfeed

A problem discovered during testing was that some feeds carried a Dutch language code while the audio contained other languages. While some feeds were manually deleted some material is maybe unaccounted for. This problem might be solved in new version of the prototype by language checking the complete feed (check for example if descriptions are Dutch) or even the speech in the podcast.

The amount of available material from the found feeds was checked from February 2007 until September 2007 (checks were performed on the 19th). During these months the spiders keep collecting new feeds and removing feeds that were no longer in use. To make sure broken or dead links to podcasts were taken into account a random download of 1000 podcasts was attempted each month. This made it possible to give a better estimate of the actual availability of podcasts. Podcasts that were successfully downloaded were checked on several characteristics such as size, duration and bitrate. During July, August and September daily checks were also performed to gather more information about day to day activity. Each day all the podcast feeds were checked for new material. If new material was found it was downloaded and analysed to collect exact information about the size and duration of each daily update.

3.3 Results

First of all, the current international supply and demand will be discussed to see the global state of podcasting. After this the results of the research on Dutch supply and demand will be presented and analysed. Then some characteristics of Dutch supply will be explored in view of the development of the prototype. Finally the future of podcasting is discussed, highlighting some potential opportunities and problems.

[1] Downloaded from: http://www.softpedia.com/get/IPOD-TOOLS/Podcast/Podspider.shtml

3.3.1 International Supply and Demand

Since the introduction of podcasting the international supply has grown continuously. Figure 3.2 illustrate this growth in the number of feeds that publish podcasts. The difference between the numbers of feeds between both websites is caused by the function of each site. Whereas FeedBurner is responsible for publishing feeds, PodcastAlley collects them. The actual number of podcasts that are now available at PodcastAlley also has grown continuously: from 30,000 podcasts back in June 2005 to more then 2.1 million in November 2007. That is an average growth of a little more then 70,000 podcasts per month.

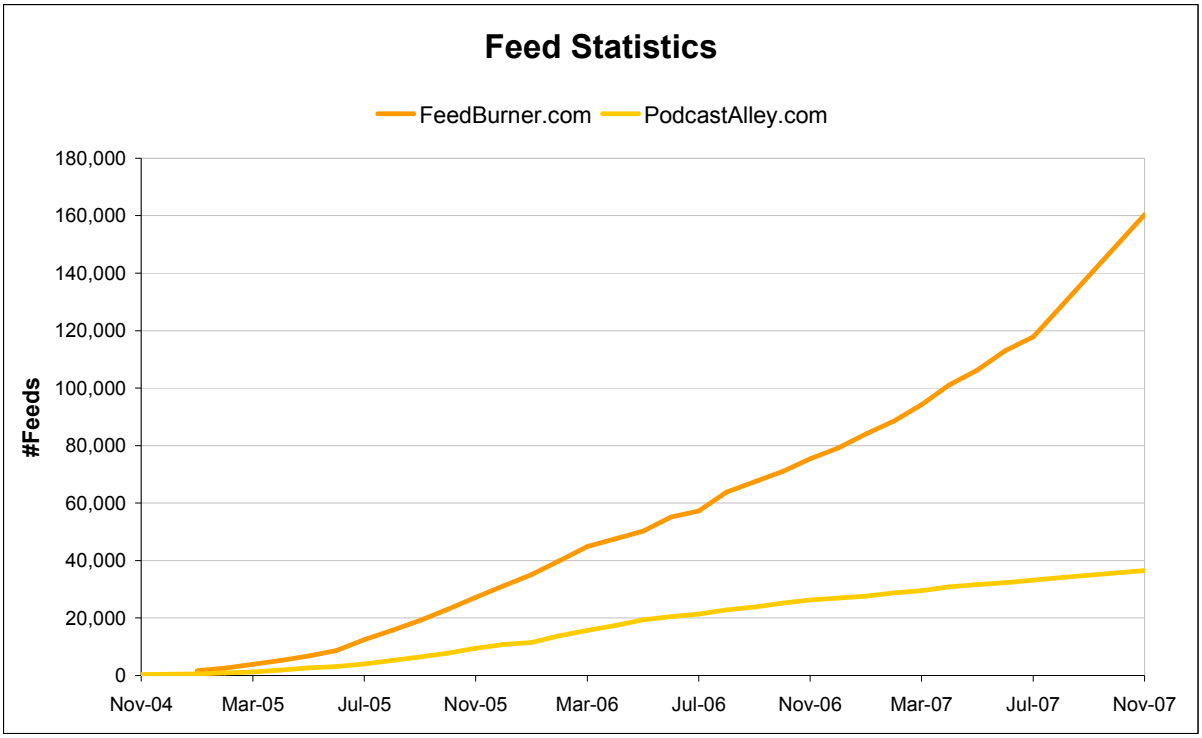


Figure 3.2: Number of feeds registered at PodcastAlley.com and Feedburner.com from Nov 2004 until November 2007.

The demand for podcasts has also grown through the years. Numbers from several sources are shown in Figure 3.3. It should be noted that the number of users is determined differently for each research:

- Research done by Arbitron/Edison Media Research in Q1, 2006 concludes that 11% (27 million) of Americans have ever listened to a podcast [6].
- Nielssen//NetRating claimed in July 2006 that about 6,6% (9.2 million) of the U.S. adult online population recently downloaded a podcast and 4,0% (5.6 million) recently downloaded a vodcast [7].
- Internet & American Life Project concluded in August 2006 that about 12% (28.2 million) of American Internet users have downloaded a podcast [8]. This was 5% more then the February-April survey.
- Statistics published by FeedBurner in December 2006 showed that there were more then 6 million aggregate subscribers, people that track shows with special software, to manage FeedBurner podcastfeeds. FeedBurner also concluded that the ratio of downloads to subscriber's average 2:1 indicating that the number of downloads is even bigger [9].

- Libsyn[1], a podcast distribution service, posted a record number of 63.4 million downloads in January 2007 [10].

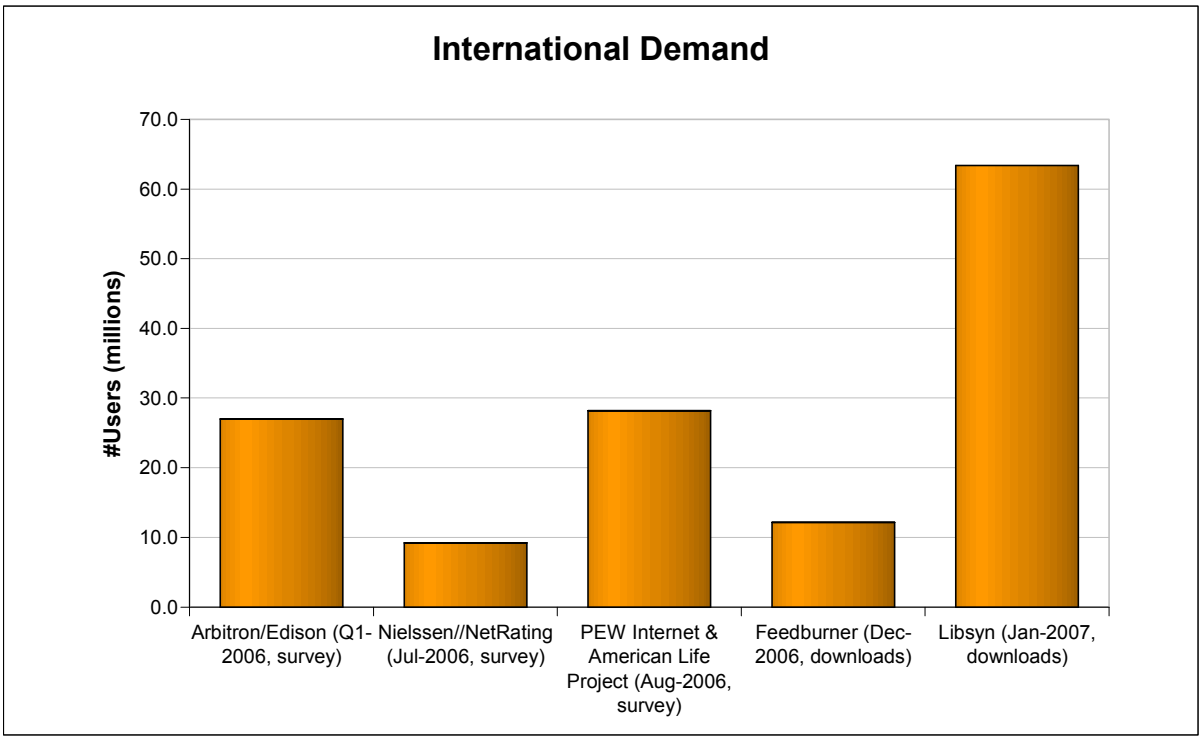


Figure 3.3: International demand as presented by different sources.

The August survey done by PEW Internet & American Life Project however confirms the figures from an older research Forrester Research in March 2006 [11] that only 1% of the people regularly download podcasts. Research performed by Yahoo in August 2005 also showed that although 28% were aware of podcasting only 2% were subscribed to a show at that time [12].

Although research shows that a large portion of the people online never or don't regularly download podcasts the figures show that podcasts are still downloaded and listened to by millions of people. The popularity of the technology is also shown by the ongoing growth of feeds and podcasts since its introduction. With respect to SDR technology it is certainly an interesting environment for development. This is also confirmed by several online SDR applications for podcasts already available.

3.3.2 Dutch Supply and Demand

Around 584 podcastfeeds and 56 vodcastfeeds were available during the research on quantity of Dutch pod- and vodcasts showing no real growth or decline. On average 7870 hours (418 GB) of Dutch podcast material and 207 hours (59 GB) of Dutch vodcast material is directly available (taken into account broken en dead links) via these Dutch feeds (see Appendix A for more information).

The amount of material offered fluctuated during the research period of eight months (see Figure 3.4). Every dip in the figure however can be explained. The dip in May was caused by the update of a Dutch radio station that reduced the amounts of items available through their feeds from 2623 items to only 83 items. The dip in September

[1] LibSyn, http://www.libsyn.com

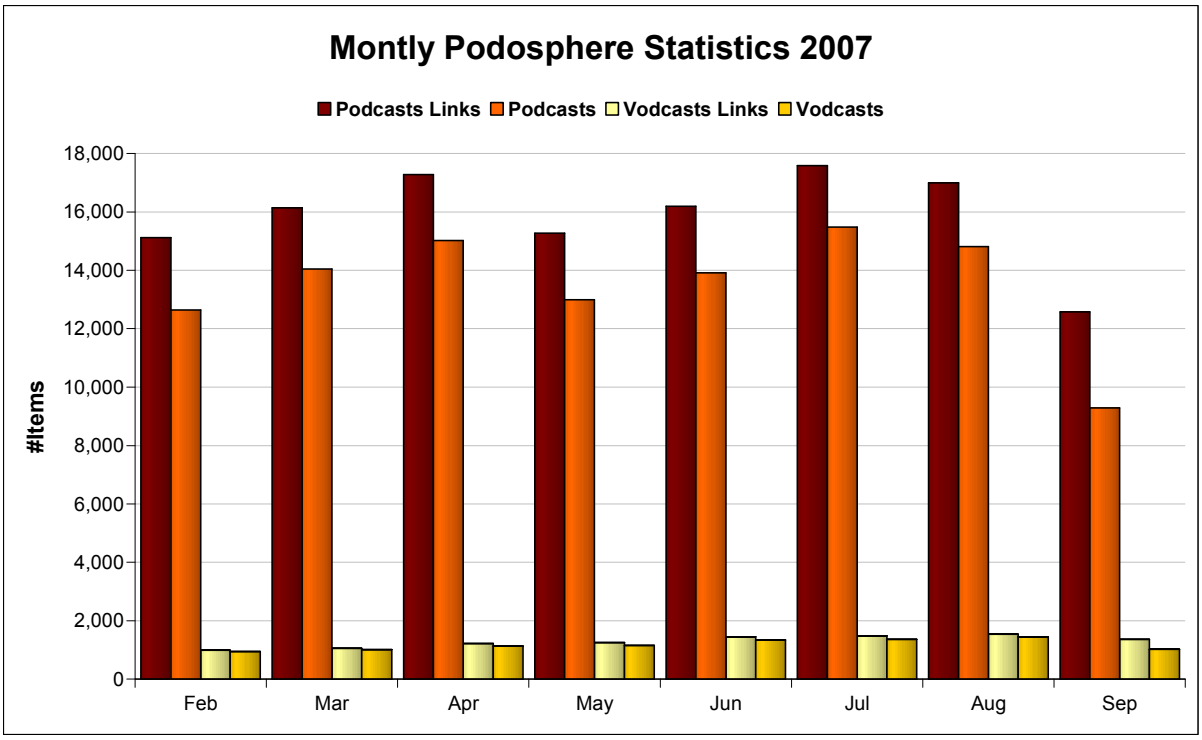


Figure 3.4: Number of links offered to pod- and vodcasts and estimation of pod- and vodcast actually available for download based on an attempt download of 1000 items.

was caused by the sudden removal of several feeds of the public radio in connection to copyright payments for music used in the podcasts. These situations are hard to foretell, which makes it difficult to predict the amount of material that will be directly available. It seems, however, that at any moment at least 10,000 podcast are directly available for download. Comparing the retrieved podcast download links with download links of the previous month showed that on average 2,746 new links were available. Also around 271 podcast feeds (46.4% of total) and 14 vodcast feeds (25.3% of total) at least published one new podcast in the first nineteen days of the month. On average 113 new podcasts with duration of 69 hours (3.7 GB) become available each day. As shown in Figure 3.5, however, the size of daily updates changes per day. Overall the direct supply and daily additions are on such a scale that human processing would take an enormous number of person hours. Automating this process would be a logical decision.

Information about the Dutch demand and number of downloads for podcasts is scarce, but research performed in the autumn of 2005 with 414 Dutch online adults up to 65 years showed that podcasting was still quite unknown [13]: About 45% of the people knew what podcasting was and about 17% of the interviewed people had ever listened to a podcast. Other people heard about it, but did not know exactly what the term meant. A pilot done at a Dutch university in 2006 showed that students have high interest in material that is made available through podcasts [14]. The experiment with 78 law students also showed that more then 78% thought other courses than used in the experiment should make material available through podcasts as well.

Unfortunately no statistics were found about the current situation. It can be assumed, since the technology has been available for an additional two years, the overall

identification and usage of podcasting has grown, but this has not been confirmed by research.

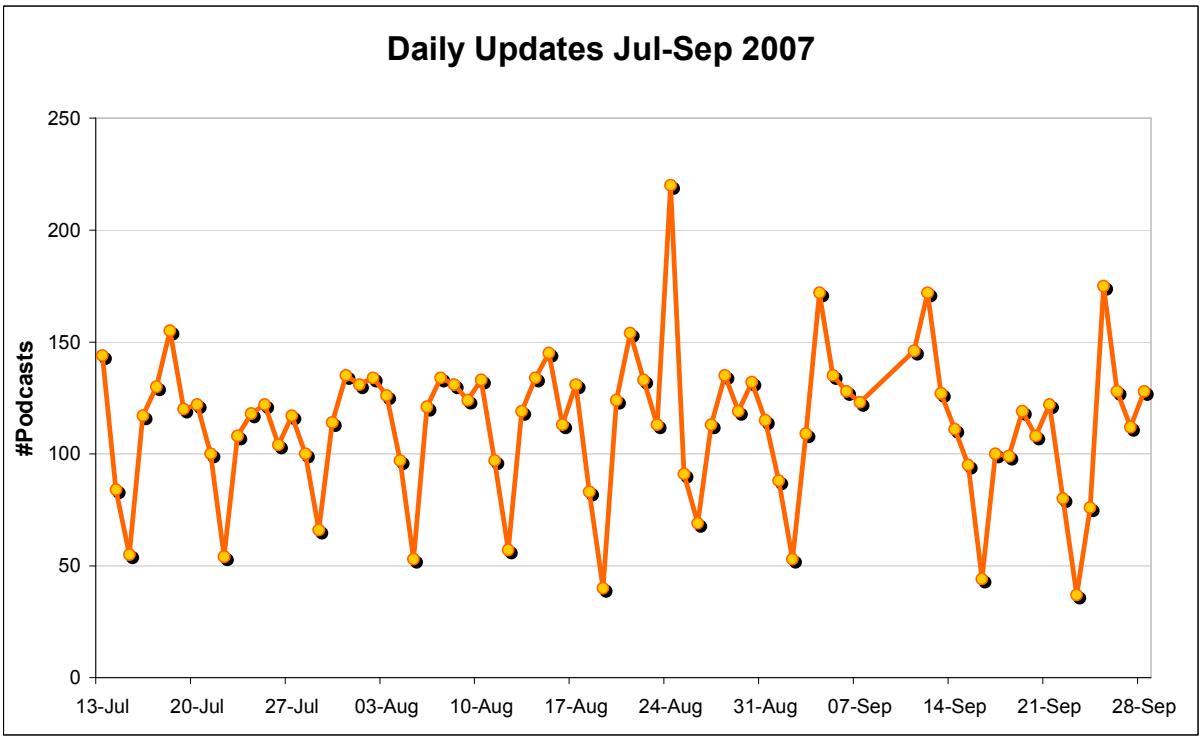


Figure 3.5: Daily amount of newly published podcasts from 13 July 2007 until 28 September 2007.

3.3.3 Characteristics of Dutch Supply

The Dutch material that was found and downloaded was checked on favourite format, average bitrate and sample rate (see Table 3.1) to see what kind of audio the SDR system should be able to handle. The characteristics are also examined because the quality of the audio has a direct influence on the automatically generated transcripts of ASR. With poor audio quality (e.g. bad recording, audio with a lot of background noise) ASR has more difficulty in recognising speech, which leads to more errors in the generated transcripts.

	Podcasts	Vodcasts
Favourite Format	.mp3 (97.4%)	.mp4/.m4v (81.5%)
Average Bitrate (kb/s)	127.0	100.5
Sample Rate 44100 Hz	90.0%	56.9%

Table 3.1: Features of Dutch Supply

Podcasts are mostly offered in mp3-format and can be considered the standard podcast-format. With an average bitrate of 127 kb/s (128 kb/s is commonly used for encoding audio) and 90% offering a sample rate of 44100Hz (equal to audio CD) the quality of recording appears to be good. It has to be taken into account however, that the quality of the audio is also dependant on other variables like environment and equipment. A noisy environment or bad microphone can decreases the quality of the audio considerably, which can result in poor quality transcripts generated by the ASR.

Vodcasts are mostly offered in mp4/m4v-format, but the mov-format (11.9%), the Quicktime player file format, is also a format used for a considerable part of the vodcasts. This would mean a system requiring to process more then 90% of available vodcast should consider supporting these two formats. The bitrate and sample rate of vodcasting are considerably lower then podcasting. This can be explained by the focus on video instead of the audio during the creation of vodcasts and the available bandwidth that has to be divided between audio and video. This could indicate ASR might be harder for video with lower quality audio, which should be taken into account when developing a system for vodcasts.

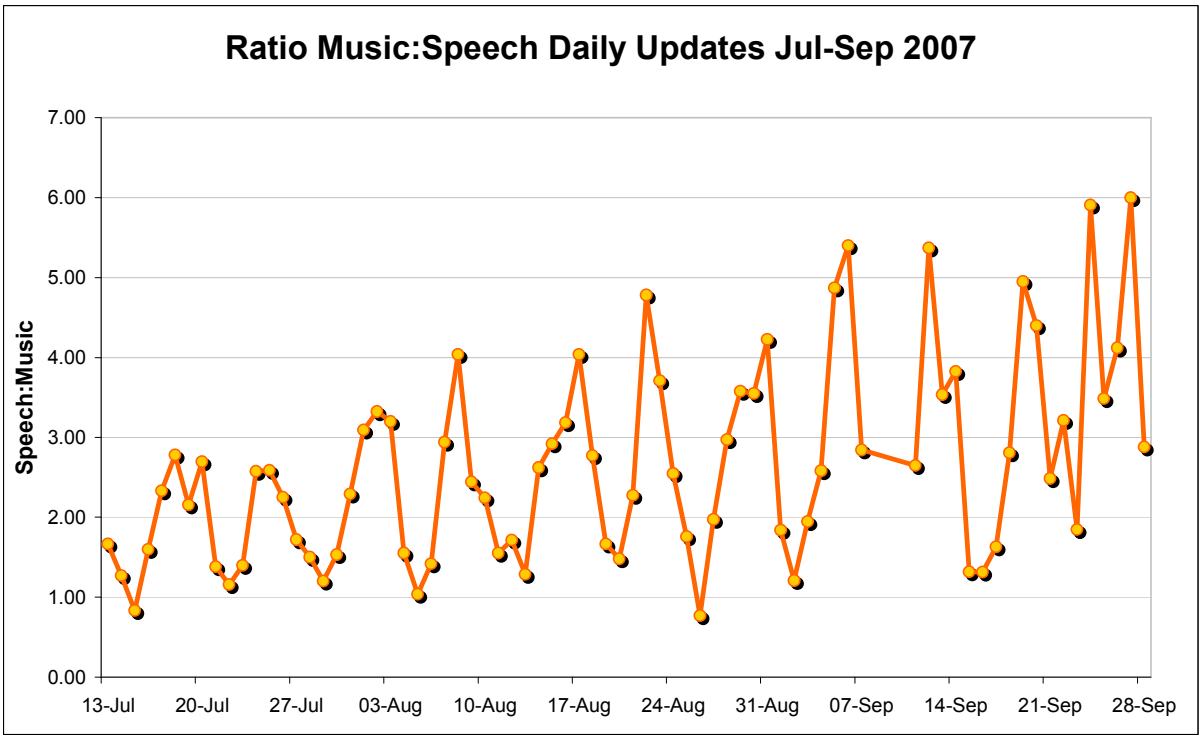


Figure 3.6: Daily ratio between speech and music podcasts.

The music-speech ratio was also researched and is relevant taking the purpose of the system into account. The focus of the system is to make information inside spoken language retrievable. In this case podcasts with little or no speech and so with a lot of music are not relevant for the system to process. To make sure the daily offered material does not only consist out of music podcasts the ratio between music and speech (for exact definition of music and speech podcast see paragraph 4.5) was monitored during July, August and September (see Figure 3.6). Using the classifier developed for the prototype (discussed in paragraph 4.5) a ratio of 2.7:1 was determined. This can be translated into a daily average of 81 speech and 31 music podcasts. As can been seen from the Figure 3.6, however, the ratio seems to be slowly rising. This could be explained by a shortcoming of the classifier, which has been trained on a static set of podcast metadata from a certain date. This means the information stored in the classifier could become less and less relevant over time when metadata of podcasts keeps changing.

3.3.4 Future of Podcasting

Research by several companies indicates that podcasting has a bright future. Forrester Research, Inc. estimated that about 1,7% (1.9 million) of the U.S. households will have adopted podcasting in 2007 growing to 12.3% (12.3 million) in

2010 [15]. The Diffusion Group predicts the podcasting user base to approach 60 million US consumers in 2010 [16]. eMarketer, Inc. estimates an active podcast audience (individuals who download one or more podcasts per week) of 7.5 million in 2008 and 15.0 million in 2010. The total podcast audience (individuals who ever downloaded a podcast) is estimated to be around 25 million people in 2008 and 50 million in 2010 [17]. A financial report of PQ Media also claims that podcasting will become an interesting advertising market [18]. While podcast advertising only totalled $3.1 million dollars in 2005 in the U.S., it is projected to reach $327.0 million in 2010.

There are also some critical notes about the future of podcasting. One of the problems with podcasting is the extra effort required to access the media. It still takes more steps to access a podcast than a newspaper or a television programme. With a part of the shows featuring the same content published by traditional media, these extra steps to download podcasts seem unnecessary, if the information is also more easily accessible from these sources. Podfading, the discontinuing of a show, is also a problem. A part of the podcasters is hobbyists having to make time to produce shows. Because of little payback or the lack of time episodes are sometimes no longer produced causing a show to slowly 'fade'. Together with the notion that podcasters are free to decide when they want to publish a new podcast a lot of uncertainty is introduced into the supply of podcasts.

3.4 Conclusion

With on average 69 hours of new Dutch material published each day and 8000 hours of Dutch podcast material directly available for download, there is a steady supply of new Dutch podcasts with an adequate amount of podcasts directly available. It has to be taken into account, however, that the directly available supply varies from day to day, since it is affected by circumstances which are difficult to predict. It seems though that at any given time more then 10,000 podcasts are immediately available. The daily addition of new podcasts also contributes to a steady supply.

With only some research available it is hard to come to any conclusion on the current and future demand of Dutch podcasts. It can be assumed that there is a demand for Dutch podcasts taking the steady supply of Dutch material into account. It is difficult to tell however if the Dutch demand is growing. While it could be assumed the demand for Dutch podcast is growing because international demand for podcasts is predicted to grow, an interesting point is the continuing growth of international feeds while the Dutch amount of feeds seems to be steady. This could indicate the demand for Dutch material has stabilized.

With 97.4% of the podcast being published in the mp3-format it can be considered the standard podcast-format. The audio quality, only taking bitrate and sample rate into account, seems sufficient for ASR. Also the 2.71 to 1 speech-music ratio supports the focus to make information inside spoken language retrievable.

Overall it can be concluded that the first sub-hypothesis -*new Dutch podcasts are regularly available expanding the podosphere in such a way human organising would take more time then automatic organising* - is partially confirmed. While new Dutch podcasts are indeed available daily expanding the podosphere it is not proven automatic organising would indeed be quicker than human organising. This would depend on the speed of an automatic system. It can be concluded, however,

automatic organising would be more logical with the amount of podcasts directly available and the continuous addition of new material.

The second sub-hypothesis -*the interest in Dutch podcasting is growing thus increasing the number of Dutch podcast downloads* - can not be confirmed. While it can be assumed there is demand for Dutch podcasts, taking the steady supply of new podcasts into account, no figures have been found to prove this assumption. In addition, no information was found about the future demand for Dutch podcasts.

In conclusion the quantity and quality of Dutch podcasts encourages the development of a SDR application. Further research however should look more into the demand for Dutch podcasts. While international demand is growing and is predicted to grow for years to come little information is available about Dutch demand. Especially the difference between the growing amount of international feeds and the stabilisation of Dutch feeds raises some questions.

4 Analysis

In this chapter the information that is available and can be extracted from podcasts is discussed. Firstly the metadata generated by the user is examined. Following this, the automatic generation of metadata for podcasts is explored and researched. Finally the performance of an automatic speech recogniser on Dutch podcasts is tested and whether podcasts would be retrievable using the speech inside the podcast.

4.1 Available Information

Information collected from podcasts can be divided into several categories. The division made in this thesis is based on the effort in terms of time to extract the information. This division was adopted taking the evaluation of the system into account: are podcast better retrievable when information from a new layer is added and is the effort extracting this information worth the performance increase? Dividing the information based on effort created three layers of information. The first layer is the user-generated metadata that is available in the feed containing the item. The second layer is the information, which comes available by analysing the file itself, such as size, duration and ID3-tag. The third layer of information is the audio itself. The first layer is readily available. The second and third layer offer extra information about the podcast, but retrieving this information requires more time because the item must be downloaded and analysed.

4.2 Information from Internet

The most commonly used standard for publishing podcasts nowadays is RSS 2.0[1]. RSS stands for *Really Simple Syndication* and can be seen as a dialect of XML. A RSS document, normally referred to as *feed*, *web feed*, or *channel*, is the first layer of information available and is directly available from the Internet. The size of these files normally ranges between 1kb and 300 kb depending on the amount of information and number of items offered in the feed. With a download speed of 1mb/s it would take less then a second to download most feeds. The RSS document normally contains detailed information about the feed and items it offers (see Figure 4.1). This information is utilized by the user to decide whether to download and listen to a podcast. This means feeds are an important source of information for users and determine a significant part of the accessibility of podcasts. Feeds are also used by podcast-clients, which automatically check the user's subscribed feeds for new content. It is important for these clients that the feeds conform to the RSS standard because the automated process is based on this standard.

Analysis of the Dutch podcastfeeds downloaded from February until September however shows that the quality of feeds in terms of information content and RSS standard is very poor. First of all about 12.5% of feeds and 15.0% of the podcasts is missing a description or even the description-tag used for describing the channel or podcast. Second 62.6% of the offered feeds do not conform to the standard and 19.7% receives warnings[2]. This shows only little attention is paid to the actual

[1] RSS 2.0 Specification (version 2.0.9) , http://www.rssboard.org/rss-specification
[2] Feedvalidator, http://feedvalidator.org

broadcast of podcasts while it is an important information source for users and podcast-clients can experience problems reading from these invalid/warned feeds.

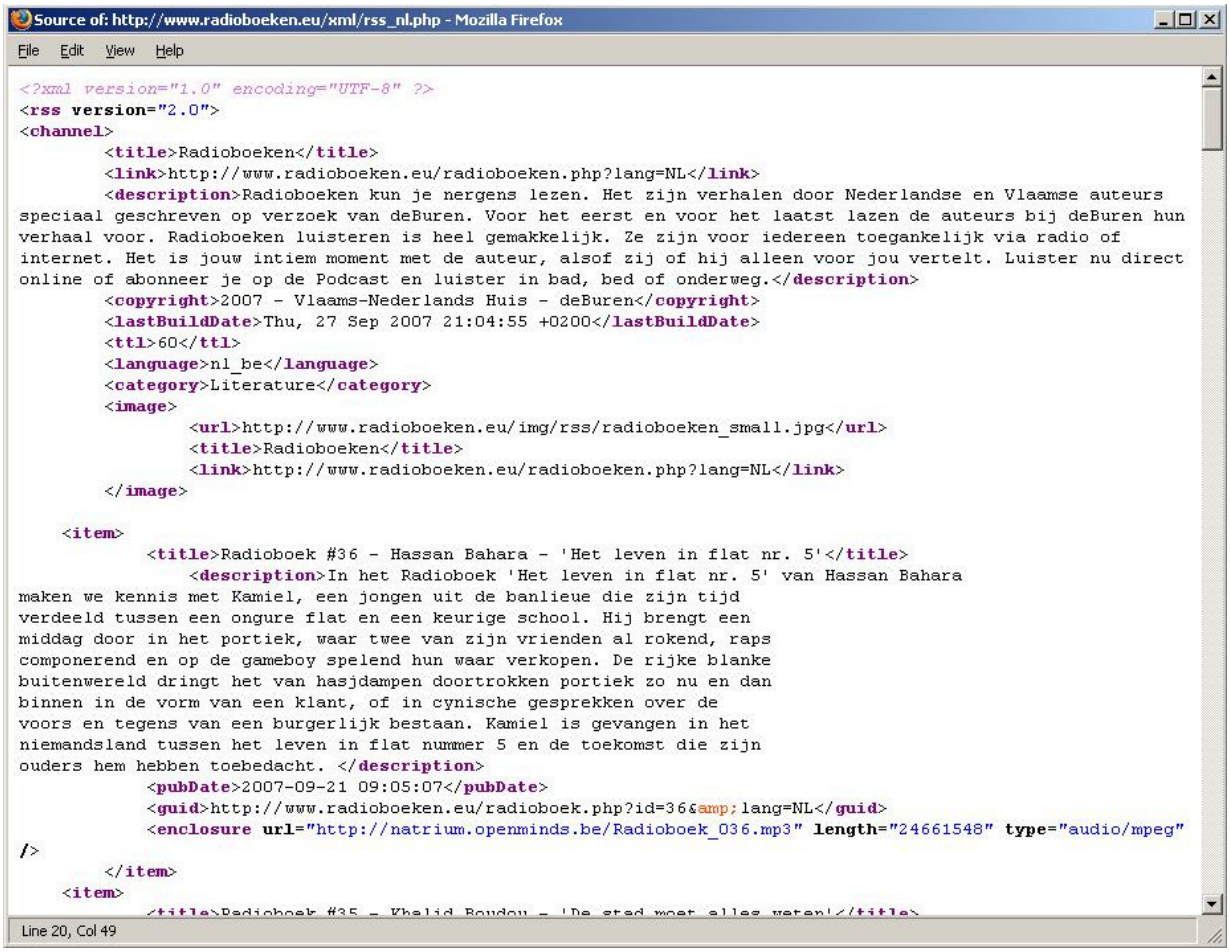


Figure 4.1: Example of podcastfeed containing a feed description and podcast descriptions.

4.3 File Information

Pod- and vodcasts are offered in a range of file formats, but with most material published in mp3-format the information that can be extracted from this file format is examined in this paragraph. Most files, however, contain two categories of information just as the mp3-format: user-generated metadata and standard file information. To obtain this second layer of information the podcast has to be downloaded. With the sizes of the podcasts normally ranging from 2mb to 60mb it would take 2 seconds to 1 minute with a 1mb/s Internet connection to download them. The information that becomes available once the file is downloaded for mp3-files can be divided in two parts:

- User-generated information inside the ID3-tag[1]. Producers and publishers however are not obligated to use the tag. The tag comes in two versions:
  - ID3v1: has a set size an only offers limited space for information. ID3v1 makes it possible to hold information about song title, artist, album, track number, year, comment and genre.
  - ID3v2: can hold a variable amount of information. ID2v2 makes it possible to hold information about song title, artist, album, track number, year,

[1] Home - ID3.org, http://www.id3.org/

comment and genre, but also provides space for more metadata such as cover art and lyrics.

- Information about the file itself such as size, duration, sample rate and bit rate.

Using the information acquired from the second layer to index podcasts gives some advantages. First of all, it becomes possible to search for podcasts based on their duration or size. Second, the extra user-generated metadata retrieved from the ID3-tags can provide more information about the podcast making it easier to retrieve.

4.4 Information from Speech

The last layer of information is the audio itself. While it is the most interesting layer it is also the most difficult layer to extract information from. While a lot of information can be extracted from audio the focus of the prototype is to decode speech inside podcasts to text with ASR. This would improve accessibility of podcasts significantly, making it possible for the users to search for podcasts based on the speech inside the podcast. Podcasts could also be classified using the generated text. This would make it for example possible for users to search through categories in the same fashion as podcast directories such as PodcastAlley. In terms of time effort a speech recogniser, depending on the complexity, takes normally up to 10 times real time to process an audio file. This would mean processing a podcast of 30 minutes could take up to 5 hours on a single computer. This process can be accelerated using more computers (processing) power.

4.4.1 Automatic Speech Recognition

In ASR the goal is to convert speech to text. While the technology has greatly improved over the years recognition accuracy varies depending on the language (state of current research and complexity of language) and domain (e.g. read speech, spontaneous speech, and background noise). Depending on the recognition accuracy, text transcriptions generated can be even used for the retrieval of speech excerpts inside the audio. The ASR system that was used during the research is a large vocabulary continuous speech recognition (LVCSR) system (see Figure 4.2).

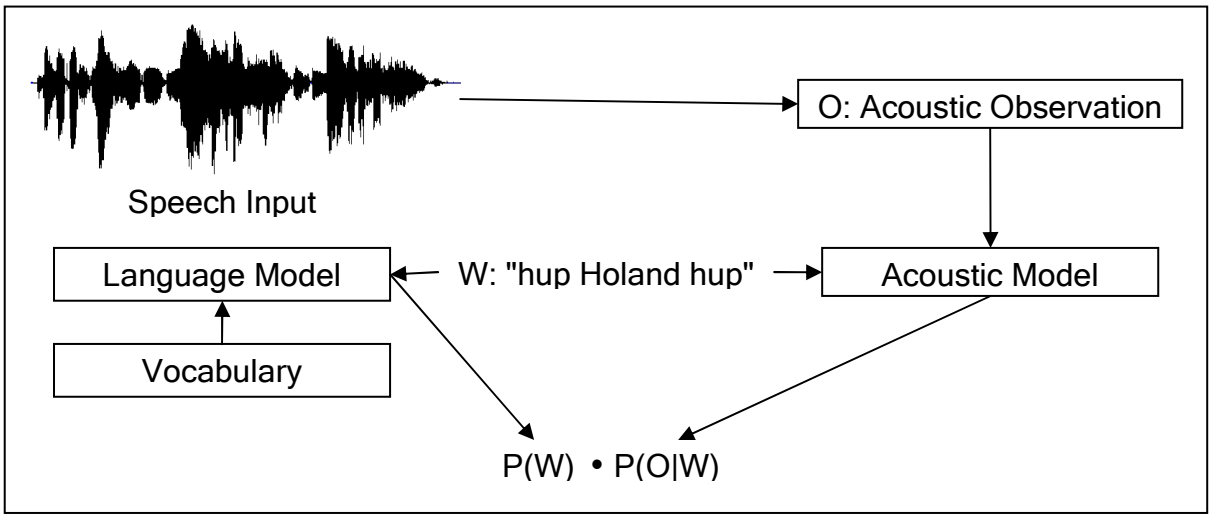


Figure 4.2 Simplified Architecture of a Large Vocabulary Continues Speech Recognizer (LVCSR)

In a LVCSR system the goal is to determine the most probable sequence of words based on the acoustic observation; a sequence of vectors with each vector

representing a digital representation of a small period of time of the speech input (typically 10 milliseconds). To select the sequence of words W with the highest probability based on acoustic observation O the probability P(W|O) is computed for each sequence.

Using Bayes' rule the probability P(W|O) can be transformed to P(W) * P(O|W) / P (O). This equation show that to find the most likely word sequence the maximum product of P(W) and P(O|W) must be found. P(W) or the *prior* is the probability of utterance W being observed independent of the perceived acoustic observation. This probability is determined by the Language Model (LM). The language model is partly based on a vocabulary that can range from a few hundred to tens of thousands of words. The dictionary used during the experiments contained roughly 51 thousand words. The number of words in the speech input that do not exist in the vocabulary are referred to out-of-vocabulary (OOV) and can be taken as quality of the speech recognition vocabulary and the language model in terms of word coverage with respect to the domain. P(O|W) is the probability of acoustic observation O or the *likelihood* given a specified word sequence W. This probability is determined by the Acoustic Model (AM).

4.4.2 Hypothesis

The Dutch podosphere is a very broad domain covering all kinds of sub-domains such as news, discussion and sport programs of varying degrees of quality. Therefore it can be assumed the performance of the speech recogniser will vary heavily depending on the podcast it has to transcribe.

The Word Error Rate (WER) is a common metric of measuring the performance of an automatic speech recogniser. The WER can be computed as:

$$WER = \frac{S + D + I}{N},$$

Where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions comparing the generated transcription and the reference text and N is the number of words in the reference.

With podcasts from all kinds of sub-domains it can also be expected that the OOV rate fluctuates depending on the kind of podcasts it has to transcribe. Taking all this into account the following hypothesis was formulated:

- *The Word Error Rate and out-of-vocabulary differs to a great extent per individual podcast.*

4.4.3 Method

To check the hypothesis the UT-BN2002 broadcast news speech recognition system was used. It is a Dutch speech recognition system developed at the University of Twente and has a WER of about 30% on broadcast news shows [19]. A set of 10 podcasts (157 minutes, see Appendix B.1-5 for more information) were first manually transcribed using Transcriber[1] and then automatically transcribed by the speech recogniser. The set consisted of several types of podcasts, but all having more

[1] Transcriber, http://trans.sourceforge.net/

speech then music in them. Sclite[1], a tool for scoring and evaluating the output of speech recognition systems, was used to determine the performance of the speech recogniser.

4.4.4 Results

Studies [20][21] have shown that adequate retrieval of speech segments from audio recordings is possible as long as the WER is below 50%. With none of the ten WERs even below 60% (see Table 4.1, more information B.6) this might suggest that the retrieval of podcast excerpts is currently not possible. It has to be taken into account, though, that the sample is too small to represent the complete podosphere. It gives, however, some indications.

The speech recognition system had an average WER of 84% on the podcasts. The best result (WER of 62.4%) was obtained on an audioblog where a man spoke about his former work experiences in Africa. The worst result (WER of 98.5%) was obtained on a radio play based on the cartoon “Tom Poes en het huilen van Urgje” by Marten Toonder. This illustrates that there is a large difference in performance of the speech recogniser between the different types of podcasts.

The results also illustrate there is still much room for improvement. A current study [22], for example, concludes that using an up-to-date language model generated with metadata and other information from the Internet can decrease the WER on podcasts by 1.4% to 10%. The out-of-vocabulary (OOV) rate of the ASR, words in the speech outside the recognition vocabulary used in the language model, has an average of 6.2% showing there is certainly room for improvement.

ID	WER	OOV
3	62,40%	4,00%
10	66,30%	6,00%
2	78,10%	5,80%
7	81,20%	6,70%
5	85,00%	7,60%
9	87,60%	5,80%
4	91,70%	4,80%
1	94,50%	6,40%
6	94,80%	7,00%
8	98,50%	7,90%

Table 4.1: ID, Word Error Rate (WER) and Out-Of-Vocabulary (OOV) rate for each podcast.

4.5 Classifier

To provide the ASR only with podcasts containing specific speech a classifier was developed to automatically divide downloaded podcasts into two categories:

- Music: podcasts with only music and podcasts with a focus on music with speech only related to the music, show filler or phone-in game.
- Speech: podcasts with only speech, podcasts with a focus on speech with music as filler or podcasts with a focus on music with speech related to interviews, news and other informative talk.

[1] NIST SCLITE Scoring Package Version 1.5, ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/sclite.htm

The ASR-module is only presented with the podcasts classified as speech.

4.5.1 Method

The classification is based on the information that is available from the feed (the metadata of the feed up till the first item and the specific metadata of the item, see Example 4.1) and the podcast itself (header-information, ID3v1-tag, ID3v2-tag retrieved from the file, see Example 4.2).

```
<channel>
        <title>boekencast.be - boek.be</title>
        <link>http://www.boekencast.be/channels_detail.php?id=4</link>
        <description>als belangenorganisatie voor het boekenvak in vlaanderen wil boek.be 'het
        boek' in al zijn verschijningsvormen koesteren en promoten. op boekencast.be wil
        boek.be een brede waaier van uitzendingen aanbieden met boekvoorstellingen, debatten,
        interviews en reacties van uitgevers, auteurs, lezers en andere
        boekenliefhebbers.</description>
        <language>nl-be</language>
        <category>literature</category>
        <image><url>http://www.boekencast.be/uploads/channels/boekbe_bc_144b.jpg</url><title>b
        oekencast.be - boek.be</title>
        <link>http://www.boekencast.be/channels_detail.php?id=4</link>
        </image><copyright>boek.be</copyright>
        <lastbuilddate>sun, 20 may 2007 23:25:32 +0200</lastbuilddate>
        <docs>http://blogs.law.harvard.edu/tech/rss</docs>
<item>
```

Example 4.1: Information from feed

```
<podcast>
        <feed>http://www.boekencast.be/rss.php?id=4</feed>
        <item>
                <title>Boek.be #11 - Peter Ghyssaert - Kleine lichamen</title>
                <description>Op de uitreiking van de Herman de Coninckprijs las Peter Ghyssaert
                het gedicht 'Omdat ik de liefde niet het standbeeld had' uit zijn genomineerde
                bundel Kleine lichamen (Querido). U hoort eerst het jurycommentaar gelezen door
                Piet Piryns en na het gedicht vertelt Peter Ghyssaert tevens waarom hij koos
                voor de titel Kleine lichamen.</description>
                <link>http://www.boekencast.be/episodes_detail.php?channel=4&amp;id=124</link>
                <guid>http://www.boekencast.be/episodes_detail.php?channel=4&amp;id=124</guid>
                <pubDate>Thu, 01 Feb 2007 11:43:07 +0100</pubDate>
                <enclosure url="http://natrium.openminds.be/boekbe_011.mp3" length="2182237"
                type="audio/mpeg"/>
        </item>
        <info>
                <size>2115709</size>
                <totaltime>132</totaltime>
                <bitrate>128</bitrate>
                <vbr>no</vbr>
                <frequency>44100</frequency>
                <channelmode>mono</channelmode>
                <copyrighted>No</copyrighted>
        </info>
        <id3v1>N/A</id3v1>
        <id3v2>
                <title>Boek.be #11 - Peter Ghyssaert - Kleine lichamen<title>
                <artist>Boek.be</artist>
                <album>Boek.be(BoekenCast.be)</album>
                <year>2007</year>
                <comment></comment>
                <track></track>
                <genre>Literature</genre>
        </id3v2>
        <classification></classification>
</podcast>
```

Example 4.2: Information from podcast

This information is then stripped from tag, punctuation and numbers. URL's are replaced with the tag URL. The rest of the information is presented to the classifier as one big string (see Example 4.3).

```
boekencast be boek be URL als belangenorganisatie voor het boekenvak in vlaanderen wil boek be
het boek in al zijn verschijningsvormen koesteren en promoten op boekencast be wil boek be een
brede waaier van uitzendingen aanbieden met boekvoorstellingen debatten interviews en reacties
van uitgevers auteurs lezers en andere boekenliefhebbers nl-be literature URL boekencast be
boek be URL boek be sun may URL URL boek be peter ghyssaert kleine lichamen op de uitreiking
van de herman de coninckprijs las peter ghyssaert het gedicht omdat ik de liefde niet het
standbeeld had uit zijn genomineerde bundel kleine lichamen querido u hoort eerst het
jurycommentaar gelezen door piet piryns en na het gedicht vertelt peter ghyssaert tevens
waarom hij koos voor de titel kleine lichamen URL URL thu feb no mono no n a boek be peter
ghyssaert kleine lichamen boek be boek be boekencast be literature
```

Example 4.3 – Generated String

A naive Bayes classifier is used for classifying because the offered string is limited in information and lacks context due to the use of multiple tags. Several studies [23] also showed that naive Bayes classifiers are very effective in text classification despite their simplistic approach. To solve the zero-probability problem Lidstone Law for smoothing is used:

$$P(w_i) = \frac{\lambda + count(w_i, c_i)}{\lambda |V| + N}$$

With $count(w_i, c_i)$ the number of times that word w_i occurs within the training documents of class c_i, $|V|$ the number of different words in documents of class c_i and N the total number of words in documents of class c_i.

The classifier was trained and tested using a dataset of 500 items, which were manually classified. These items were randomly downloaded from 487 valid feeds (offering 12,699 items in total) on the 20th of May 2007. The performance (correctly classified items) of the classifier using different values for lambda was deduced taking the average of 100 runs. Every run the dataset was randomly divided in 80% training data and 20% test data.

4.5.2 Results

Using a lambda of 0.2, as shown in Table 4.2, gave the best performance of the classifier.

Λ	Performance	Remark
1.0	88.49%	Also known as add-one smoothing or Laplace Law
0.9	88.54%	
0.8	88.68%	
0.7	88.86%	
0.6	89.10%	
0.5	89.24%	Also known as Jeffreys-Perks Law
0.4	89.48%	
0.3	90.14%	
0.2	**90.37%**	
0.1	90.19%	

Table 4.2: Average performance of the naïve Bayes classifier on 100 runs using different values for lambda in Lidstone Law for smoothing.

Training the classifier with the full 500 items and using a lambda of 0.2 it was able to correctly classify 90% of a new set of 200 random unseen items (downloaded from the same 487 valid feeds). The classifier currently used by PodVinder has been trained with these 700 items.

4.6 Conclusion

The available information of a podcast can be divided into three layers taking the effort in terms of extraction time as the criterion for division: online, file and speech. Each of these layers has advantages and disadvantages.

First of all the online layer is easy to collect, but research has shown that the quality of the user-generated metadata is poor with some feeds and podcast even missing a description at all.

The second layer itself is divided in two parts; user-generated information and file information. The file information in particular offers advantages. This makes it possible to search for podcasts of a certain size or duration. The downside of collecting this information is the necessity to download the podcast, which can take a good amount of time depending on the size and download speed.

The last layer is the speech layer. This is the most important layer because it contains the information the user is interested in, but retrieving this information is difficult. Although it is possible to use ASR to convert the speech to text, accuracy varies depending on the domain. Using a Dutch speech recognizer specialised in broadcast news on 10 podcasts gave an average WER of 84%. Based on the research performed partial retrieval of podcasts is not possible. Previous research shows adequate retrieval of parts of speech inside audio is possible as long as the WER is below 50%. It has to be taken into account, however, that the set of 10 podcasts is too small to represent the complete podosphere

Current research also shows that the WER can decrease using an up-to-date language model generated with metadata and other information from the Internet. The average OOV rate (6.2%) on the 10 podcasts shows there is certainly room for improvement in the language model.

Overall it can be concluded that the hypothesis - *The Word Error Rate and out-of-vocabulary differs to a great extent per individual podcast* - is confirmed. Even with a small set of 10 podcasts the difference in WER is around 35% between the best and worst generated transcript. Also the OOV show quite a difference, with 4.0% as the lowest rate and 7.9% as the highest rate.

5 Search

In this chapter the actual retrieval of podcasts is discussed. First a short introduction is given on information retrieval, search engines and the evaluation of search engines. Following this, the first evaluation performed of the prototype, which was developed during the thesis, is discussed and analysed.

5.1 Information Retrieval

Information retrieval is the process of retrieving information based on an information need. Looking up a telephone number from the phonebook to call somebody is a very simple example of this. In the context of computer science, information retrieval is the automatic retrieval of (pieces) of documents within a large collection based on a query representing the information need of a user. The key goal of an information retrieval system is presenting the user with results that might be useful or relevant.

5.2 Search Engines

A search engine is commonly used as name for an information retrieval system designed to help find information stored on one or more computer systems. The main goal of a search engine is to maximize precision, the fraction of the documents retrieved that are relevant to the user's information need, and recall, the fraction of the documents that are relevant to the query that are successfully retrieved, and to minimize the response time. The most commonly known and used search engines are web search engines such as Google and Yahoo. During September 2007 these two search engines handled 7.6 billion American searches[1].

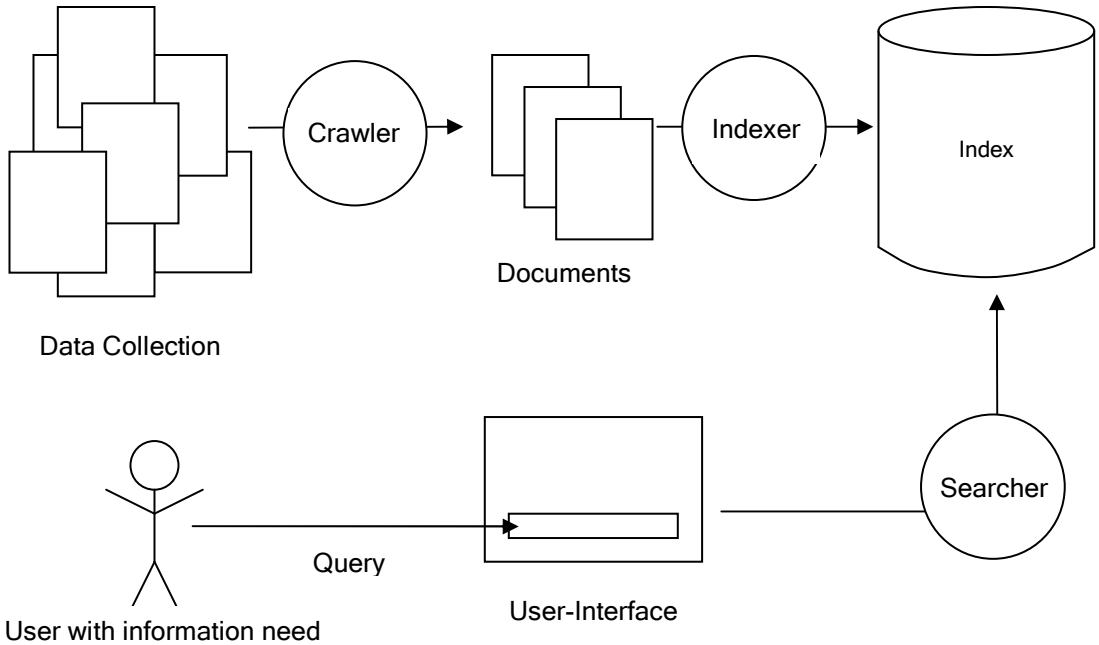


Figure 5.1: Basic architecture of an information retrieval system

[1] comScore Releases September U.S. Search Engine Rankings, http://www.comscore.com/press/release.asp?press=1805

The basic architecture of a search engine (see Figure 5.1) consists of five parts: crawler, indexer, index, searcher and a user-interface. One or more crawlers go through a data collection and deliver documents to the indexer to be stored into the index. The indexer indexes the found documents using the information available. A user can then give the system a query, a representation of his information need, with the user-interface. The searcher transforms the query back into an information need, collects documents that fulfil this information need and returns a list of documents ranked on usefulness and relevancy if any are available.

Spoken Document Retrieval (SDR) systems are search engines that specialise in the retrieval of excerpts from speech recordings or complete speech recordings. While most of the basic architecture of search engines is the same, the major difference is the usage of automatic speech recognition between crawler and indexer to decode the documents (speech recordings) into text, giving the indexer information to index.

5.3 SDR Evaluation

Tague-Sutcliffe formulates evaluation of retrieval systems as follows [24]:

Evaluation of retrieval systems is concerned with how well the system is satisfying users not just in individual cases, but collectively, for all actual and potential users in the community. The purpose of evaluation is to lead to improvements in the information retrieval process, both at a particular installation and more generally.

The Text REtrieval Conference (TREC) is a conference aimed at supporting and encouraging evaluation and research within the information retrieval community. It has run evaluations for information retrieval systems for over a decade now. The first SDR evaluation was at TREC-6 in 1997. The evaluation of several SDR systems was done that year by a so called known-item retrieval task which simulates a user seeking a particular, half-remembered document in a collection. The goal in a known-item retrieval task is to recover a single correct document for each topic. From 1998 ad-hoc retrieval tasks were used. In an ad-hoc task a list of documents ranked by decreasing similarity to the topic is returned. The returned documents are then evaluated on relevance by a team of human assessors making it possible to compute various evaluation scores.

5.4 Hypothesis

Research has shown that the better the performance of a speech recognition system, the better the retrieval performance will be [20][25]. This would imply that the better the transcription (lower Word-Error-Rate) the easier it will be to retrieve a podcast. Another logical assumption is that an item is easier to retrieve if more information is available about that item. This implies indexing podcasts based on the metadata from the Internet plus extra metadata extracted from analysing the podcasts should result in better retrievability. To determine whether user-generated metadata from the ID3-tags, which requires downloading the files, improves retrievability an extra hypothesis was added.

While an ad-hoc task has more resemblance to normal usage of a search engine, it has a complex topic selection process and needs expensive human relevance assessments. Given the available time and resources it was decided to perform a known-item evaluation.

In order to verify whether the retrievability of podcasts is improved the Mean Reciprocal Rank (MRR), a metric to compare different retrieval methods, is calculated. The MRR is the mean of the reciprocal of the rank at which the known item was found averaged over all items, using 0 as the reciprocal for queries that did not retrieve the item.

Combining the assumption and retrieval task the three following hypothesis were formulated:

- *Podcasts with a higher Word Error Rate are harder to retrieve than podcasts with a lower Word Error Rate.*
- *Indexing Dutch podcasts using the extra user-generated metadata retrieved from ID3-tags results in better retrievability (higher MRR) than indexing Dutch podcasts with only user-generated metadata available from the Internet.*
- *Indexing Dutch podcasts with transcripts generated by ASR results in better retrievability (higher MRR) than indexing Dutch podcasts with only user-generated metadata*

5.5 Retrievability of Podcasts

To test the first hypothesis and see whether there is a correlation between the accuracy of the automatic generated transcript and retrievability, some of the data and results from the ASR evaluation in paragraph 4.4.3 were used.

5.5.1 Method

A random set of 60 podcasts, classified as speech (total duration 1285 minutes), from a total of 14,234 podcasts was downloaded on the 11th of September 2007. These podcasts were automatically transcribed by the speech recogniser. Then the manual and automatically generated transcripts of the 10 podcasts used for the ASR evaluation in paragraph 4.4.3 were used to create two different indexes (one with 60 + 10 automatically generated transcripts and one with 60 automatically generated transcripts + 10 manually created transcripts). Two retrieval runs were performed on each index using first the title and then title and description available from the feed as queries for retrieval of the 10 podcasts.

5.5.2 Results

As expected and shown in Table 5.1 the retrievability of a podcast is better when the manually generated transcript is used instead of the transcript generated by ASR. It seems, however, that the level of the WER isn't directly related to the retrievability of an item. A podcast with a very high WER can still be retrieved at rank one while another podcast with a lower WER might not be retrieved at all.

These results can be explained by a combination of two factors. Firstly, the provided information in the feed (title and description) of certain podcasts provides better material for queries than others. Podcasts 4 and 9 aren't retrieved taking the title as query because they are both completely QOV (OOV words that were used in the query). In addition, the query based on the title of podcast 5 consists of 2 words, one of which was recognisable. Second, a clear introduction and/or ending of a podcast can also improve retrievability. This is for example the case with the podcast 8 having the highest WER while still retrieved at rank 1. The podcast (radio play) is retrieved based on a combination of an informative title and description and clearly spoken

summary at the beginning or end of the episode that was well recognised by the ASR.

		Title			Title & Description		
ID	WER	#Words	QOV	Ranking ASR (Manual)	#Words	QOV	Ranking ASR (Manual)
3	62,40%	9	33%	1 (1)	40	13%	1 (1)
10	66,30%	4	25%	Not Retrieved (1)	31	39%	2 (1)
2	78,10%	4	25%	2 (1)	8	25%	2 (1)
7	81,20%	3	33%	1 (1)	51	24%	1 (1)
5	85,00%	2	50%	Not Retrieved (4)	6	83%	Not Retrieved (5)
9	87,60%	1	100%	Not Retrieved (Not Retrieved)	44	20%	9 (1)
4	91,70%	1	100%	Not Retrieved (Not Retrieved)	14	36%	1 (Not Retrieved)
1	94,50%	9	22%	8 (1)	64	16%	1 (1)
6	94,80%	12	58%	32 (1)	104	26%	13 (1)
8	98,50%	9	22%	1 (1)	51	24%	1 (1)

Table 5.1: ID, Word Error Rate (WER), the number of words in the query, the QOV (OOV words in the query) and retrieval rank of podcasts using title or title and description, available from the feed, as query in a set of 70 podcasts indexed on transcripts generated by ASR or manually generated transcripts.

A noteworthy point is the retrieval of podcast 4. While it is not retrieved based on the manually generated transcript it is retrieved based on the transcript of the ASR. Further examination showed that due to a recognition error the transcript contained a word that was not uttered. This word, however, caused the podcast to be retrieved.

5.6 SDR Evaluation

SDR is successfully applied to podcasts in English and Spanish for fragment retrieval to increase accessibility (Everyzing[1] and Podscope[2]). In this experiment, however, the first aim is to research whether extra generated metadata from analysis of the file can also improve full podcast retrieval. This decision was based on the results from chapter 4 where it was concluded that segment retrieval currently might not be possible because of high WER levels.

5.6.1 Data

The same 60 randomly download podcasts, as described in paragraph 5.5.1, were used. To gather more information about the retrievability of certain types of podcasts, they were split into three categories based on the motivation of the publisher.

- Business: podcasts published and produced by groups (e.g., political parties, businesses, cultural organisations) using podcasting as a new means of promotion.
- Non-professional: podcasts published and produced by people with personal interest in the medium as a form of communication.
- Professional: podcast published and produced by traditional broadcast channels using podcasting as a new publishing platform.

[1] Everyzing, http://www.everyzing.com
[2] Podscope, http://www.podscope.com

From each category, 20 podcasts were downloaded and processed by the ASR-module. Topics were generated by students for 30 of the podcasts (10 for each category). The students were asked to create a title and a small description (maximum of one sentence) of the podcasts based on the content of the podcast with no extra information or metadata given.

5.6.2 Indexing and Retrieval

For indexing and retrieval the standard libraries plus Dutch analysis classes of Lucene were used[1]. Due to limited time no research was performed in improving search results by modifying these standard libraries. All the metadata that was collected and generated (feed, file and ASR) was combined and indexed into one single field using the standard Indexer of Lucene. Queries were processed by the standard QueryParser. Also for retrieval and scoring the standard classes of Lucene were used. Lucene uses a combination of the Boolean mode and the Vector Space Model (VSM) of information retrieval to determine how relevant a given document is to a user's query. Firstly, the boolean model is used to narrow down the documents that need to be scored based on the use of boolean logic. Then the VSM is used to score the document. The idea behind the VSM is the more times a query term appears in a document relative to the number of times the term appears in all the documents in the collection, the more relevant that document is to the query[2].

The score of query q with terms t for document d correlates to the cosine-distance or dot-product between document and query vectors in a Vector Space Model (VSM) of Information Retrieval. A document whose vector is closer to the query vector in that model is scored higher. The score is computed as follows[3]:

$$score(q,d) = coord(q,d) \bullet queryNorm(q) \bullet \sum_{t \in q} (tf(t \in d) \bullet idf(t)^2 \bullet t.getBoost() \bullet norm(t,d))$$

where:

- $tf(t \in d)$ correlates to the term's frequency, defined as the number of times term t appears in the currently scored document d. Documents that have more occurrences of a given term receive a higher score.
- $idf(t)^2$ stands for Inverse Document Frequency. This value correlates to the inverse of the number of documents in which the term t appears. This means rarer terms give higher contribution to the total score.
- $coord(q,d)$ is a score factor based on how many of the query terms are found in the specified document. Typically, a document that contains more of the query's terms will receive a higher score than another document with fewer query terms.
- $queryNorm(q)$ is a normalizing factor used to make scores between queries comparable. This factor does not affect document ranking (since all ranked documents are multiplied by the same factor), but rather just attempts to make scores from different queries (or even different indexes) comparable.
- $t.getBoost()$ is a search time boost of term t in the query q as specified in the query text or as set by application calls.

[1] Lucene, http://lucene.apache.org
[2] Apache Lucene - Scoring, http://lucene.apache.org/java/docs/scoring.html
[3] Simalarity - http://lucene.zones.apache.org:8080/hudson/job/Lucene-Nightly/javadoc/org/apache/lucene/search/Similarity.html

- $norm(t,d)$ encapsulates a few (indexing time) boost and length factors.

5.6.3 Runs

The created titles and descriptions were used as queries for retrieval given the following evaluation runs:

- User-generated metadata available from Internet (USG Internet): a retrieval run performed on the index generated only from the user-generated metadata available on the Internet.
- All User-generated metadata available (USG Complete): a retrieval run performed on the index generated from the user-generated metadata available on the Internet plus user-generated metadata retrieved from the ID3-tags.
- All information available (USG Complete + ASR): a retrieval run performed on the index generated from all available user-generated metadata plus transcripts generated by ASR.
- ASR transcriptions (ASR): a retrieval run performed on the index generated only from the transcripts generated by ASR.

Each run was performed twice: using the created title as query and then using the created title and description as query.

5.6.4 Measures

Performance of the system was measured based on the given ranks to the known items that were the targets of the searches. The following measures were collected:

- Mean rank when found (MRWF): the mean rank at which the known item was found averaged across all the topics.
- Mean reciprocal rank (MRR): the mean of the reciprocal of the rank at which the known item was found averaged across all the topics, using 0 as the reciprocal for queries that did not retrieve the item.
- Number of known items found on rank 1.
- Number of known items not found.

5.7 Results

Table 5.2 (using only title as query) and 5.3 (using title and description as query) show the main results of the evaluation runs (for all results see Appendix C). The USG Complete + ASR run perform the best. Only the MRWF of this run (when using title and description as query) is slightly lower then USG Complete run and USG Internet run, which is caused by the retrieval of two podcasts at low ranks not retrieved in these two runs.

Run (title as query)	MRR	MRWF	Rank 1	Not Retrieved
USG Complete + ASR	0.59	1.38	15 (50%)	9 (30%)
USG Complete	0.56	2.33	14 (47%)	9 (30%)
USG Internet	0.51	2.42	13 (43%)	11 (37%)
ASR	0.07	2.50	1 (3.3%)	26 (87%)

Table 5.2: MRR, MRWF, number of podcasts retrieved at rank 1 and number of podcasts not retrieved given 30 queries (using title) for podcasts representation based on complete user-generated metadata plus transcripts generated by ASR, complete user-generated metadata, user-generated metadata available from the Internet and transcripts generated by ASR.

Run (title & description as query)	MRR	MRWF	Rank 1	Not Retrieved
USG Complete + ASR	0.75	2.41	21 (70%)	3 (10%)
USG Complete	0.71	2.16	20 (67%)	5 (17%)
USG Internet	0.66	2.29	18 (60%)	6 (20%)
ASR	0.22	8.67	5 (17%)	15 (50%)

Table 5.3: MRR, MRWF, number of podcasts retrieved at rank 1 and number of podcasts not retrieved given 30 queries (using title and description) for podcasts representation based on complete user-generated metadata plus transcripts generated by ASR, complete user-generated metadata, user-generated metadata available from the Internet and transcripts generated by ASR.

Both tables also show that using the extra information which becomes available from downloading the podcast has a positive effect on every measure higher in the USG Complete Run than in the USG Internet Run.

5.7.1 Missing Transcripts

Nine out of the thirty podcasts that had a topic created were fully recognised as music by the ASR (but classified as speech by the classifier) and did not contain a transcript. The absence of these transcripts, however, did not negatively influence results. This can be explained by other podcasts receiving a lower score due to the inclusion of a transcript (resulting in less relevance to the topic when scored by the VSM) while the podcasts without a transcript received the same score, thus ranking them higher. This happened in 2 cases using title as query and 1 case using the title and description as query.

The experiment was repeated omitting all the podcasts without transcripts in the whole set of 60 podcasts (this left a set of 43 podcasts with 21 topic podcasts) to see how big the missing ASR transcripts influence was on the measures. This yielded the results shown in Table 5.4 and 5.5. As expected with a smaller set, the MRR and MRWF increased in all runs and the percentage of podcasts retrieved and podcast retrieved at rank 1 improved. Overall, however, no noteworthy differences were found.

Run (title as query)	MRR	MRWF	Rank 1	Not Retrieved
USG Complete + ASR	0.64	1.20	12 (57%)	6 (29%)
USG Complete	0.62	1.27	11 (52%)	6 (29%)
USG Internet	0.55	1.23	10 (48%)	8 (38%)
ASR	0.13	2.25	2 (10%)	17 (81%)

Table 5.4: MRR, MRWF, number of podcasts retrieved at rank 1 and number of podcasts not retrieved given 21 queries (using title) for podcasts representation based on complete user-generated metadata plus transcripts generated by ASR, complete user-generated metadata, user-generated metadata available from the Internet and transcripts generated by ASR.

Run (title and description as query)	MRR	MRWF	Rank 1	Not Retrieved
USG Complete + ASR	0.84	2.38	16 (76%)	0 (0%)
USG Complete	0.78	1.63	15 (71%)	2 (10%)
USG Internet	0.69	1.72	12 (57%)	3 (14%)
ASR	0.38	6.87	7 (33%)	6 (29%)

Table 5.5: MRR, MRWF, number of podcasts retrieved at rank 1 and number of podcasts not retrieved given 21 queries (using title and description) for podcasts representation based on complete user-generated metadata plus transcripts generated by ASR, complete user-generated metadata, user-generated metadata available from the Internet and transcripts generated by ASR.

5.7.2 Retrieval on ASR Transcripts

Looking at the results of the ASR run in Table 5.5 it can be seen that 15 out of the 21 podcasts (71%) with transcripts were retrieved based on speech extracted from the podcast. From the 15 podcast that were retrieved only 5 had an overall positive effect on the retrieved rank when all the metadata was used (USG Complete + ASR run). This was mainly because 9 podcasts were already retrieved at rank 1 only using the provided user-generated metadata from the Internet. This means the retrievability of 5 of the 6 podcasts, which were not retrieved at rank 1, improved using the transcript generated by the ASR. The ranking of 3 podcasts respectively increased 1, 2 and 7 positions. Two podcasts, although retrieved at low rankings (12 and 17 respectively), weren't retrieved at all using the user-generated metadata.

Investigation of the scoring showed that in the case of the 5 podcasts ASR transcripts mainly contained words that were not present in the user-generated metadata. This increased the number of words that matched the terms in the query resulting in a relatively higher score then the other documents once the ASR transcript was included.

Table 5.4, however, shows that the query plays an important role when making use of the extra information provided by the transcript of the podcast. From the 21 podcasts with a transcription, only 4 (19%) were retrieved when only the title was used as query. With 2 of these items already retrieved at rank 1 the inclusion of the transcript increased the ranking of 1 item while decreasing the rank of another podcast.

5.7.3 Professional, Non-Professional and Business

When comparing the three different categories on the full set of 60 podcasts, professional podcasts seem the best retrievable in terms of MRR and MRWF (see table 5.6). Non-professional podcasts however are all retrieved, while two professional podcasts are not found. The difference between the categories however is minimal when one considers that the MRR and MRWF of the business category are only lowered by the retrieval of two podcasts on low ranks.

USG Complete + ASR (title & description as query)	MRR	MRWF	Rank 1	Not Retrieved	No ASR
Professional	0.80	1.00	8 (80%)	2 (20%)	4
Non-Professional	0.78	2.00	7 (70%)	0 (0%)	3
Business	0.66	4.11	6 (60%)	1 (10%)	2

Table 5.6: MRR, MRWF, number of podcasts retrieved at rank 1, number of podcasts not retrieved given 30 queries (using title and description) and number of podcast without transcription for podcasts representation based on complete user-generated metadata plus transcripts generated by ASR categorised by professional, non-professional and business.

5.7.4 User-Generated Metadata

A big difference can be found, however, when comparing the different categories using the user-generated metadata for retrieval (see table 5.7). While the MRR in the categories professional and non-professional are respectively 0.65 and 0.71 the MRR of the business category is 0.31. This is caused by only five of the ten podcasts being retrieved. This indicates that the user-generated metadata of business podcasts is of low quality content-wise.

USG Complete (title as query)	MRR	MRWF	Rank 1	Not Retrieved
Non-Professional	0.71	2.22	6 (60%)	1 (10%)
Professional	0.65	1.14	6 (60%)	3 (30%)
Business	0.31	4.20	2 (20%)	5 (50%)

Table 5.7: MRR, MRWF, number of podcasts retrieved at rank 1, number of podcasts not retrieved given 30 queries (using title) and number of podcast without transcription for podcasts representation based on complete user-generated metadata categorised by professional, non-professional and business.

Another difference appears (see Table 5.8) in the MRR and not found of the USG Internet run using only the created titles as query. This could indicate that user-generated metadata available from the Internet for non-professional podcasts is of very good quality content-wise.

USG Internet (title as query)	MRR	MRWF	Rank 1	Not Retrieved
Non-Professional	0.71	2.33	6 (60%)	1 (10%)
Professional	0.50	1.00	5 (50%)	5 (50%)
Business	0.31	4.00	2 (20%)	5 (50%)

Table 5.8: MRR, MRWF, number of podcasts retrieved at rank 1 and number of podcasts not retrieved given 30 queries (using title) for podcasts representation based on user-generated metadata from the Internet generated by ASR categorised by professional, non-professional and business.

5.8 Conclusion

Research conducted revealed that the level of the WER can't directly be related to the retrievability of the podcast: the first hypothesis - *Podcasts with a higher Word Error Rate are harder to retrieve than podcasts with a lower Word Error Rate.* - is not confirmed. Although a WER can be very high, a clearly spoken and informative summary at the beginning or end of a podcast can contain enough information, which can be extracted by the ASR, to facilitate full podcast retrieval.

The results of the SDR experiments look promising with the MRR increasing when using information extracted from speech inside the podcast. Also using the extra user-generated information for indexing, which comes available when downloading the podcast, shows a positive effect on retrievability. The experiments, however, show that the influence indexing podcasts with the transcript generated by the ASR is currently limited. This was especially the case when using small queries. The ranking of only 1 podcast increased, while the ranking of 1 podcast decreased. The results were better using more descriptive queries: 5 of the 21 rankings improved. Investigation showed that the ranking mainly improved because the ASR contained words that were not present in the user-generated metadata. This increased the number of words that matched the terms in the query resulting in a relatively higher score then the other documents once the ASR transcript was included

When comparing the three categories using all the user-generated metadata and transcript from the ASR no big differences are shown in retrievability. The MRR and MRWF are considerably lower of the business category, but this is only caused by the retrieval of two podcasts on low ranks. The differences are bigger though when looking at the user-generated metadata. Using the title as query on all user-generated metadata the MRR of the business category is noticeably lower then the other two categories. This indicates user-generated metadata of this category is low quality content-wise. In contrast, when using the title as query on user-generated metadata from the Internet, the MRR of the non-professional category is noticeably

higher then the other 2 categories. This indicates that the user-generated metadata provided by non-professionals is of very good quality content-wise.

Overall it can be concluded that both the first - *Indexing Dutch podcasts using the extra user-generated metadata retrieved from ID3-tags results in better retrievability (higher MRR) than indexing Dutch podcasts with only user-generated metadata available from the Internet* - and second hypothesis - *Indexing Dutch podcasts with transcripts generated by ASR results in better retrievability (higher MRR) than indexing Dutch podcasts with only user-generated metadata* - are confirmed based on the results from this experiment.

6 Conclusion and Future Work

Combining the results obtained throughout this research gives some interesting insights into the feasibility of speech-based retrieval of Dutch podcasts in terms of supply, demand and technology. Most of the research, however, was performed on a small scale and only gives some first indications. These indications, nonetheless, are positive. In addition, parts of the results were also obtained on a first prototype, which has significant room for improvement.

6.1 Collection

The development of an SDR system for Dutch podcasts is certainly worthwhile considering the amount of material that is offered. More then 10,000 podcasts are directly available for download and daily updates average 113 podcasts, with a duration of 69 hours. It can be concluded that the Dutch podosphere is growing, with a considerable foundation already available. The demand for Dutch podcasts, however, is unknown. While the international demand is predicted to grow, it is difficult to conclude likewise in the Dutch case. Especially since the offer of Dutch podcastfeeds has stabilized, while the offer of international podcast feeds continues to grow. It can be assumed, however, that there is at least a steady demand for Dutch podcasts considering the constant supply of new podcasts. It can be concluded that the second part of the main hypothesis - *Current and future supply of and demand for Dutch podcasts validates this approach* - is partly confirmed.

The first recommendation for future research is to collect more information about the demand for Dutch podcasts. While the development of a system is attractive from a research point of view, it must be considered whether the development of a full-blown system is valuable from a user point of view.

6.2 Analysis

Results of the research performed on the first prototype also encourage the development of a SDR system for Dutch podcasts. While the current Dutch speech recogniser used doesn't yet have the accuracy to provide segment retrieval of podcasts, the information extracted increases the retrievability of complete podcasts. Bearing in mind that the speech recogniser used is specialised in broadcast news and has not been modified for this particular task, a lot of improvement can still be made. Current research shows, for example, that the WER decreases using an up-to-date language model generated with metadata and other information from the Internet.

The second recommendation for further research is the investigation of whether the performance of the Dutch speech recogniser can indeed be improved using the user-generated metadata and metadata from other sources. Improving the performance of the recogniser could improve full podcast retrieval and might make the retrieval of excerpts of podcasts possible, thus improving accessibility of podcasts even further. Users would be able to retrieve the part of the podcast which interests them most.

6.3 Search

Overall it can be concluded that the first part of the main hypothesis - *Dutch podcasts can be retrieved based on a combination of information extracted from speech inside*

the podcast and user-generated metadata. - is confirmed. In a known-item retrieval experiment 90% of the podcasts were retrieved with 78% of these podcasts at rank one using a detailed query. Using a standard query 70% of the podcasts were retrieved with 71% of these podcasts at rank one. These results show that retrieval based on user-generated metadata and information extracted from speech is indeed possible.

Another notable finding is that the accuracy of a transcript cannot be directly correlated to the retrievability of a podcasts. Podcasts with high WERs don't necessarily have to be less retrievable than podcasts that have lower WERs. Research showed, however, that the retrievability of a podcast improves when the WER decreases.

The last recommendation for further research is to investigate whether retrievability of podcasts can be improved by adjusting the manner how podcasts are indexed and retrieved. Due to limited time and resources no research was performed on this part of the prototype. Lucene, however, offers a great deal of possibilities to fine-tune indexing and retrieval of documents.

7 References

[1] Ben Hammersley, Audible Revolution, The Guardian, 12/2/2004.

[2] D. Gregoire, How to handle getting past episodes?, http://tech.groups.yahoo.com/group/ipodder-dev/message/41, 16/9/2004, 20/02/2007.

[3] Wikipedia.org, History of podcasting, http://en.wikipedia.org/wiki/History_of_podcasting, 20/02/2007.

[4] Wikipedia.org, Podcast, http://en.wikipedia.org/wiki/Podcast, 19/11/2006, 20/02/2007.

[5] J. Morang, F.M.G. de Jong, R.J.F. Ordelman and A.J. van Hessen InfoLink: analysis of Dutch broadcast news and cross-media browsing, in Proceedings of CBMI 2005, ISBN not assigned, 2005.

[6] S. Bausch and H. Leilani, Podcasting Gains an Important Foothold Among U.S. Adults Online Population, According to Nielsen//NetRatings, NetRatings, Inc., 2006.

[7] B. Rose and J. Lenski, Internet and Multimedia 2006: On-Demand Media Explodes, Arbitron Inc. / Edison Media Research, 2006.

[8] M. Madden, 12% of Internet Users Have Downloaded a Podcast, Pew Internet & American Life Project, 2006.

[9] R. Klau, Expanding Universe: Podcasting Market, Update, http://blogs.feedburner.com/feedburner/archives/2006/04/expanding_unive_1.php, 18/04/2006, 19/01/2006.

[10] Libsyn.com, Libsyn Network Posts Record Number of Downloads, http://soundoff.libsyn.com/index.php?post_id=179487, 07/02/2007, 15/03/2007.

[11] C. Li, T. Schadler, R. Fiorentino, T. McHarg, Podcasting Hits The Charts, Forrester Research, Inc., 28/03/2006.

[12] J. Grossnickle, T. Board, B. Pickens, M. Bellmont, RSS–Crossing into the Mainstream, Yahoo!/Ipsos Insight, October 2005.

[13] C. Aalberts, Podcasting en Politiek, http://www.chrisaalberts.nl/site/pages/project/archiefpodcast.htm, 2005.

[14] Podcast-proef dik geslaagd, Mare Leids Universitair Weekblad, volume 2005-2006, number 25, 23/03/2006.

[15] Schadler, Ted, Bernoff, Josh and McHarg, Tenley, The Future of Digital Audio, Forrester Research, Inc., 21/03/2005.

[16] TDG Research, Podcasting Users to Approach 60 Million US Consumers by 2010, TDG Press Releases, 15/06/2005.

[17] M. Chapman, Podcasting: Who's Tuning In?, eMarketer, Inc., March 2006.

[18] PQ Media: custom media research, Alternative Media Research Series I: Blog, Podcast and RSS Advertising Outlook, Executive Summary, PQ Media, April 2006.

[19] Roeland Ordelman, Dutch Speech Recognition in Multimedia Information Retrieval, Ph.D. thesis, University of Twente, The Netherlands, October 2003.

[20] J.S. Garofolo, C.G.P. Auzanne, and E.M Voorhees. The TREC SDR Track: A Success Story. In Eighth Text Retrieval Conference, pages 107-129, Washington, 2000.

[21] Effect of Recognition Errors on Information Retrieval Performance, Alessandro Vinciarelli, IDIAP Research Institute.

[22] Jun Ogata, Masataka Goto, and Kouichirou Eto, Automatic Transcription for a Web 2.0 Service to Search Podcasts, INTERSPEECH 2007.

[23] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, Tom Mitchell, Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, Volume 39, Issue 2-3 (May-June 2000), Pages: 103 - 134, 2000, ISSN:0885-6125.

[24] Jean M. Tague-Sutcliffe, Some perspectives on the evaluation of information retrieval systems, Journal of the American Society for Information Science, Volume 47, Issue 1 (January 1996), 1 - 3, 1996, ISSN:0002-8231.

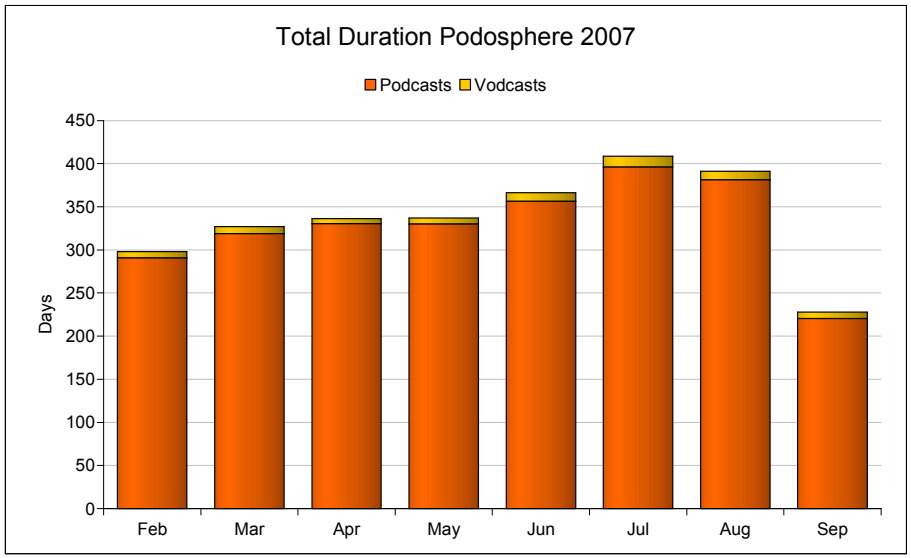
[25] [S.E. Johnson, P. Jourlin, K. Spärck Jones, and P.C. Woodland. Spoken Document Retrieval for TREC-9 at Cambridge University. In Proceedings of TREC-9, 2000.

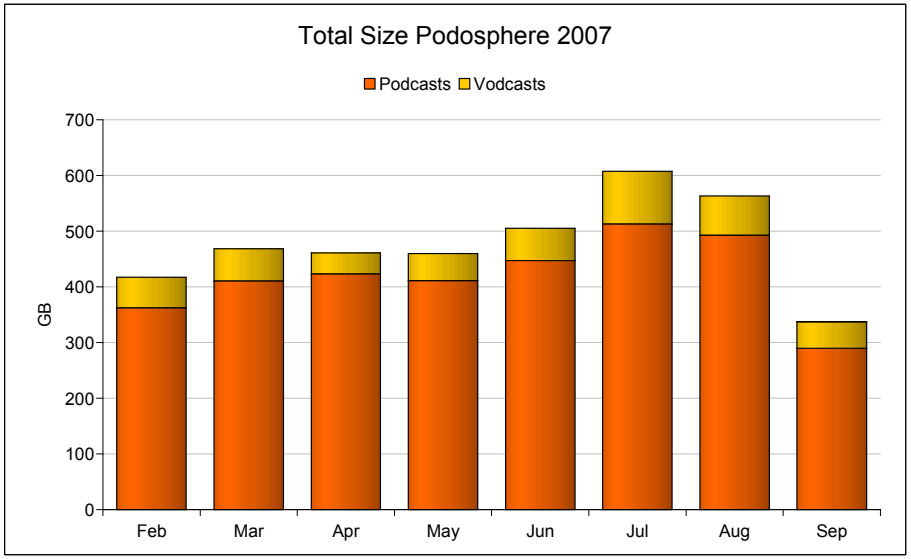
8 Appendix A – Podcast Statistics

8.1 Website List Spider

- Podfeed: http://www.podfeed.nl
- PodcastInfo: http://www.podcastinfo.nl
- PodPlaza: http://www.podplaza.nl
- Podcasting.be: http://www.podcasting.be
- Radiocast.nl: http://www.radiocast.nl
- Podcasting: http://www.podstart.nl

8.2 Monthly Statistics Podcasts & Vodcasts





8.3 Monthly Difference offered URLs

	February (15364)	March (16499)	April (17879)	May (15961)	June (17125)	July (18115)	August (17929)	September (13514)
February	-	-	-	-	-	-	-	-
March	2486	-	-	-	-	-	-	-
April	4464	3115	-	-	-	-	-	-
May	4975	3963	2063	-	-	-	-	-
June	8042	7264	5844	4924	-	-	-	-
July	9410	8690	7227	6539	2771	-	-	-
August	10027	9362	8128	7474	3861	2354	-	-
September	7636	7320	6352	6007	3232	2223	1510	-

9 Appendix B - ASR Evaluation

9.1 File Location

#	File
1	http://ictroddels.nl/audio//roddels_127.mp3
2	http://download.omroep.nl/teleacnot/hoezo-podcast/hzactua/actua20070605.mp3
3	http://www.podplaza.nl/mediafiles/145249/method_rss/Kenya_Hola_nr_133_Naar_de_tandarts_in_Nairobi.mp3
4	http://www.podplaza.nl/mediafiles/145209/method_rss/DOA_131.mp3
5	http://www.postbankdownload.nl/podcast/postbank_sectorbeleggen_podcast_mei_2007.mp3
6	http://natrium.openminds.be/Radioboek_027.mp3
7	http://www.podplaza.nl/mediafiles/145395/method_rss/Plaza_36.mp3
8	http://feeds.feedburner.com/~r/bommel/~3/123907047/091.mp3
9	http://feeds.feedburner.com/~r/Edukast/~5/123670550/Edukast-233-20070610.mp3
10	http://feeds.readspeaker.com/app/podcaster/nlreleased/audio/207/902737973.mp3

9.2 Feed Location

#	Feed
1	http://ictroddels.nl/?feed=podcast
2	http://www.rvu.nl/podcast.php?feed=hoezo_actueel
3	http://feeds.podplaza.nl/mzee
4	http://feeds.podplaza.nl/davidonair
5	http://feeds.feedburner.com/sectorbeleggenpodcast/
6	http://www.radioboeken.be/xml/rss_nl.php
7	http://feeds.podplaza.nl/plaza
8	http://feeds.feedburner.com/bommel
9	http://feeds.feedburner.com/Edukast
10	http://feeds.readspeaker.com/app/podcaster/redirect/feed/207.xml

9.3 Feed Description

#	Feed Description
1	Iedere dinsdag het ICT Nieuws als podcast en veel interviews door de week heen
2	*[EMPTY]*
3	Ervaringen die ik heb opgedaan tijdens mijn werk in Kenya, waar ik in het Tana River gebied geholpen heb bij het opzetten van een landbouwbedrijf.
4	David presenteert een show vol met nieuws, muziek en veel onzin!
5	Maandelijks op een handige manier even bijgepraat worden over beleggen in sectoren. Dat is de Sectorbeleggen Podcast van Postbank. Onze beleggingsexperts bespreken voor u hun maandelijkse Sectorvisie. In 'De wereld in 10 sectoren' krijgt u een toelichting op het advies van Postbank voor elke sector. En elke maand wordt een sector in meer detail behandeld in 'Sector in de diepte'. Interessant voor elke belegger, dus niet alleen voor klanten

	van de Postbank. Kijk voor meer informatie over Sectorbeleggen bij de Postbank op www.postbank.nl/beleggen. We wensen u veel luisterplezier!
6	Radioboeken kun je nergens lezen. Het zijn verhalen door Nederlandse en Vlaamse auteurs speciaal geschreven op verzoek van deBuren. Voor het eerst en voor het laatst lazen de auteurs bij deBuren hun verhaal voor. Radioboeken luisteren is heel gemakkelijk. Ze zijn voor iedereen toegankelijk via radio of Internet. Het is jouw intiem moment met de auteur, alsof zij of hij alleen voor jou vertelt. Luister nu direct online of abonneer je op de Podcast en luister in bad, bed of onderweg.
7	De eigen show van Podplaza, bedoeld als baken in de woelige wateren van het Podcasten. Wat is leuk, wat is nieuw, wat mag je niet missen en wie maakt het? Plaza houdt je op de hoogte! Shownotes vind je op http://plaza.podplaza.nl
8	NPS - Bommel
9	Een Educatieve Podcast. Dit is de MP3-feed voor de EduKast een Nederlandstalige podcast over ICT en Onderwijs. Op http://www.EduKast.nl/ vind je alle afleveringen en de shownotes.
10	NU.nl Rich Site Summary

9.4 Podcast Description

#	**Podcast Description**
1	Vanuit een hotellobby in Amsterdam een ICT Roddels met daarin natuurlijk het nieuws van de bijna fusie tussen Microsoft en Yahoo, kraken van MacBooks in slaapstand (de MacLockPick), de teleurgestelde Consumentenbond die het met KPN Intenter Plus Bellen niet meer ziet zit en HP. Dat laatste bedrijf mag zich namelijk voor de rechter verantwoorden voor [...]
2	Nederlands leren met spraakherkenning
3	Soms komt iets vervelends ook weer goed gelegen zoals een bezoek aan de tandarts in de hoofdstad. Na de pijn van de kies het plezier van het stadsleven na maanden bush
4	n00bcasting! David praat over imago vs. identiteit en bespreekt het podcasten in Nederland.
5	<img src="http://feeds.feedburner.com/~r/sectorbeleggenpodcast/~4/117648507" height="1" width="1">;
6	Erik Vlaminck (1954) is roman- en theaterschrijver. Vlaminck, nazaat van de beroemde Vlaamse schilder Maurice de Vlaminck, is op zoek gegaan naar herinneringen aan zijn ouders, grootouders, overgrootouders, van vaders- en moederszijde. Met die herinneringen heeft hij een moza?ek, een intrigerende roman fleuve die uit zes deelromans bestaat, gebouwd. In zijn Radioboek *Het groot huisvuil en de buren* doet een oudere man op onbedoeld komische wijze zijn beklag over zijn buren, zijn vrouw en de rest van de wereld tegen een onbekende gesprekspartner. Het venijn zit in dit verhaal in de staart.
7	Deze keer veel nieuws (waaronder over een show met 5 miljoen luisteraars en het boek In Europa als Podcast), heerlijke Podsafe rock van The Clementines en natuurlijk weer luistertips voor gamers, Smurfen, hittesters en mensen die NIET in Doetinchem wonen. Dat en nog veel meer in Plaza 36.
8	De opvoeding van Urgje baart heel wat zorgen. Heer Bommel leidt e.e.a. in juiste banen: "Het was een verwend ventje en zo groot, als je begrijpt wat ik bedoel. Maar ik heb hem met vaste hand toch klein gekregen". Verteller: Nettie Blanken
9	Aflevering 233, weer een codebash achter de rug, hoeveel hebben we er nog nodig, de opening van het Fontys eiland nadert met rappe schreden en natuurlijk aandacht voor croquet. Niet alleen ben ik verkouden vandaag (tja, het is er het weer voor), m...
10	AMSTERDAM - Het computerspel Manhunt 2 mag in Groot-Brittannië niet verkocht worden. Volgens de Britse filmkeuring BBFC, die ook videogames beoordeelt, is het spel te sadistisch. MP3

9.5 Podcast Information

#	Type	#Persons	Comments	Duration (sec.)	Bit Rate	Sample Rate
1	Information program - ICT	2		800	128	44100
2	Informative telephone interview	2		278	112	44100
3	Storytelling (audioblog)	1		365	128	44100
4	Traditional radio – Amateur	2	Contains music	2104	128	44100
5	Informative program - Stock	2		749	128	44100
6	Storytelling (book)	1	Belgian	1443	128	44100
7	Informative program – Podcasting	1	Contains music, plays 2 fragments from other podcast	1173	128	44100
8	Radio play	5+	O.B. Bommel and Tom Poes	885	128	44100
9	Informative program - Education and ICT	1	Contains music	1525	96	22050
10	Informative program – Internet	1	Text-to-speak (readspeaker)	101	48	22050

9.6 ASR Results

#	Ref. words	Hyp. words	Aligned words	Total Error (WER)	Correct	Substitution	Deletions	Insertions	Word Accuracy
1	2360	2159	2645	94.5%	17.6%	61.8%	20.6%	12.1%	5.5%
2	773	717	822	78.1%	28.2%	58.2%	13.6%	6.3%	21.9%
3	1109	1022	1155	62.4%	41.7%	46.3%	12.0%	4.1%	37.6%
4	2177	1551	2314	91.7%	14.6%	50.3%	35.0%	6.3%	8.3%
5	2405	2002	2660	85.0%	25.6%	47.1%	27.4%	10.6%	15.0%
6	3515	3165	3963	94.8%	17.9%	59.4%	22.7%	12.7%	5.2%
7	2197	1929	2440	81.2%	29.9%	46.9%	23.3%	11.1%	18.8%
8	1929	1762	2146	98.5%	12.8%	67.3%	19.9%	11.2%	1.5%
9	3319	2814	3645	87.6%	22.3%	52.7%	25.0%	9.8%	12.4%
10	243	247	267	66.3%	43.6%	48.1%	8.2%	9.9%	33.7%
Average	2002.7	1736.8	2205.7	**84.01%**	25.42%	53.81%	20.77%	9.41%	15.99%

10 Appendix C - Information Retrieval Evaluation

10.1 Ranking Individual Items

	Title as query					Title & Description as query				
	USG I	**USG C**	**USG C + ASR**	**ASR**		**USG I**	**USG C**	**USG C + ASR**	**ASR**	**Comment**
b1	0	0	0	0	b1	0	0	12	11	
b2	1	1	2	4	b2	1	1	1	6	
b3	0	0	0	0	b3	0	0	17	16	
b4	2	2	3	0	b4	2	2	2	11	
b5	0	0	0	0	b5	1	1	1	1	
b6	14	15	3	0	b6	1	1	1	0	NO ASR
b7	2	2	1	1	b7	1	1	1	1	
b8	1	1	1	2	b8	8	8	1	1	
b9	0	0	0	0	b9	1	1	1	5	
b10	0	0	0	0	b10	0	0	0	0	NO ASR
n1	2	2	1	0	n1	2	2	1	1	
n2	1	1	1	0	n2	1	1	1	0	
n3	0	0	0	0	n3	5	4	3	11	
n4	1	1	1	0	n4	1	1	1	30	
n5	1	1	1	0	n5	2	1	4	0	
n6	1	1	1	0	n6	1	1	1	0	NO ASR
n7	1	1	2	0	n7	1	1	1	1	
n8	11	10	2	0	n8	18	18	6	0	NO ASR
n9	2	2	2	0	n9	1	1	1	0	
n10	1	1	1	0	n10	1	1	1	0	NO ASR
p1	1	1	1	1	p1	1	1	1	1	
p2	0	1	1	0	p2	0	1	1	0	
p3	0	2	1	0	p3	1	1	1	0	
p4	1	1	1	0	p4	1	1	1	0	
p5	1	1	1	0	p5	1	1	1	6	
p6	1	1	1	0	p6	1	1	1	0	NO ASR
p7	1	1	1	0	p7	1	1	1	0	NO ASR
p8	0	0	0	0	p8	1	1	1	1	
p9	0	0	0	0	p9	0	0	0	0	NO ASR
p10	0	0	0	0	p10	0	0	0	0	NO ASR

10.2 Calculated Measures

Title as query

Total	USG I	USG C	USG C + ASR	ASR
MRWF	2.42	2.33	1.38	2.00
MRR	0.51	0.56	**0.59**	0.09
#1	13	14	15	2
NF	11	9	9	26

Business	USG I	USG C	USG C + ASR	ASR
MRWF	4.00	4.20	2.00	2.33
MRR	0.31	0.31	**0.32**	0.18
#1	2	2	2	1
NF	5	5	5	7

Non-Profressional	USG I	USG C	USG C + ASR	ASR
MRWF	2.33	2.22	1.33	-
MRR	0.71	0.71	**0.75**	-
#1	6	6	6	0
NF	1	1	1	10

Professional	USG I	USG C	USG C + ASR	ASR
MRWF	1.00	1.14	1.00	1.00
MRR	0.50	0.65	**0.70**	0.10
#1	5	6	7	1
NF	5	3	3	9

Title & description as query

Total	USG I	USG C	USG C + ASR	ASR
MRWF	2.29	2.16	2.41	6.87
MRR	0.66	0.71	**0.75**	0.26
#1	18	20	21	7
NF	6	5	3	15

Business	USG I	USG C	USG C + ASR	ASR
MRWF	2.14	2.14	4.11	6.50
MRR	0.56	0.56	**0.66**	0.36
#1	5	5	6	3
NF	3	3	1	2

Non-Profressional	USG I	USG C	USG C + ASR	ASR
MRWF	3.30	3.10	2.00	10.75
MRR	0.73	**0.78**	**0.78**	0.21
#1	6	7	7	2
NF	0	0	0	6

Professional	USG I	USG C	USG C + ASR	ASR
MRWF	1.00	1.00	1.00	2.67
MRR	0.70	**0.80**	**0.80**	0.22
#1	7	8	8	2
NF	3	2	2	7

11 Appendix D – Technical Documentation Collection

The first part of the system has to perform three basic tasks. First of all the Internet has to be searched for Dutch podcastfeeds. Podcastfeeds normally have a channel description including the type of language spoken in the podcast. As shown in Figure 11.1 feeds normally carry a <language>-tag. Inside this tag a language code is placed. For the Dutch language the following codes are used: nl (Dutch), nl-nl (Netherlands-Dutch) and nl-be (Vlamisch). The second task is to check the feed for updates. Checking the feed for the first time all the items can be seen as updates. Once checked for initials items the feed has to be checked regularly if new podcasts are published. This can be done by comparing the links to the podcasts offered the last time, available from the url-attribute inside the <enclosure>-tag (see Figure 11.1), with the newly available links.

Figure 11.1: Example of Dutch Podcastfeed

The last task is the actual download of the podcasts so they can be analysed in the next part of the system. To summarise the first part of the system has to perform the three following basic tasks:

- Search for valid podcastfeeds on the Internet.
- Check podcastfeeds for updates.
- Download updates.

Taking the scope of the project and these basic tasks into account several possibilities concerning the development of the prototype had to be considered.

Issue 1: What programming language will be used?
Decision: With a lot of online work and text that has to be processed the decision was made to use Perl[1]. Perl offers an easy package for Internet access and download of files. Also using regular expressions to process text is fairly easy and offers a lot of possibilities.

Issue 2: What qualifies as a valid feed?

[1] Perl.com: The Source for Perl, http://www.perl.com

Decision: With the focus on Dutch podcasts, only feeds that carry a <language>-tag with a Dutch language codes are accepted. The feed also has to provide a direct link to one or more podcasts to make sure the feed actually offers material.

Issue 3: How are feeds checked for updates?
Decision: If the document has been modified since the last check the feed is downloaded. All the links to podcasts in the downloaded feed will be retrieved and compared to the last urls known from the feed. New urls will then be saved. This requires the system to save the published urls seen in each feed at the moment.

Issue 4: How will the last urls of each feed be saved?
Decision: Each feed will have its own url-file keeping track of the last urls published in the feed.

Issue 5: In which format will the podcastfeeds metadata be saved?
Decision: Metadata will be saved in XML-format to create flexibility. This makes it possible to also handle this data easily when other applications are developed. Each feed will have its own metadata-file keeping track of the user-generated title-, link- and description-tag and url of the feed.

Issue 6: Which podcast file-formats will be supported?
Decision: Support for the mp3-format will be developed first, considering that more then 90% of the offered pod- and vodcast material is in this format.

Issue 7: In which format will the podcast metadata be saved?
Decision: Metadata will be saved in XML-format to create flexibility. This makes it possible to also handle this data easily when other applications are developed. The structure of the XML will provide space for: url of the feed that provided the podcast, user-generated metadata and information that is collected during the analysis phase.

11.1 Requirements

The requirements are based on the first three basic tasks and issues discussed. Also the scope of the project, answering a part of the research question and to serve as a foundation for further research, is taken into account.

11.1.1 Functional Requirements

1. Search Internet for Dutch podcastfeeds
2. Check if already in system (url or content)
3. Analyse if feed has language-tag with a Dutch language code
4. Analyse if feed has title-, link- an description-tag
5. Analyse if feed has enclosure-tag with mp3-file within item-tag
6. Analyse if feed only offers mp3-format
7. Check feeds for updates
8. Download file

11.1.2 Non-Functional Requirements

1. Modular

11.2 Implementation

To create flexibility the system architecture consists out modules (see Figure 11.2) that each has their own part in performing the basic tasks. The output is saved in a directory structure to create easy access for modules or even other applications build to use the generated information. The output-handling and directory structure also makes it possible to easily monitor and analyse the system (see Table 11.1 for details).

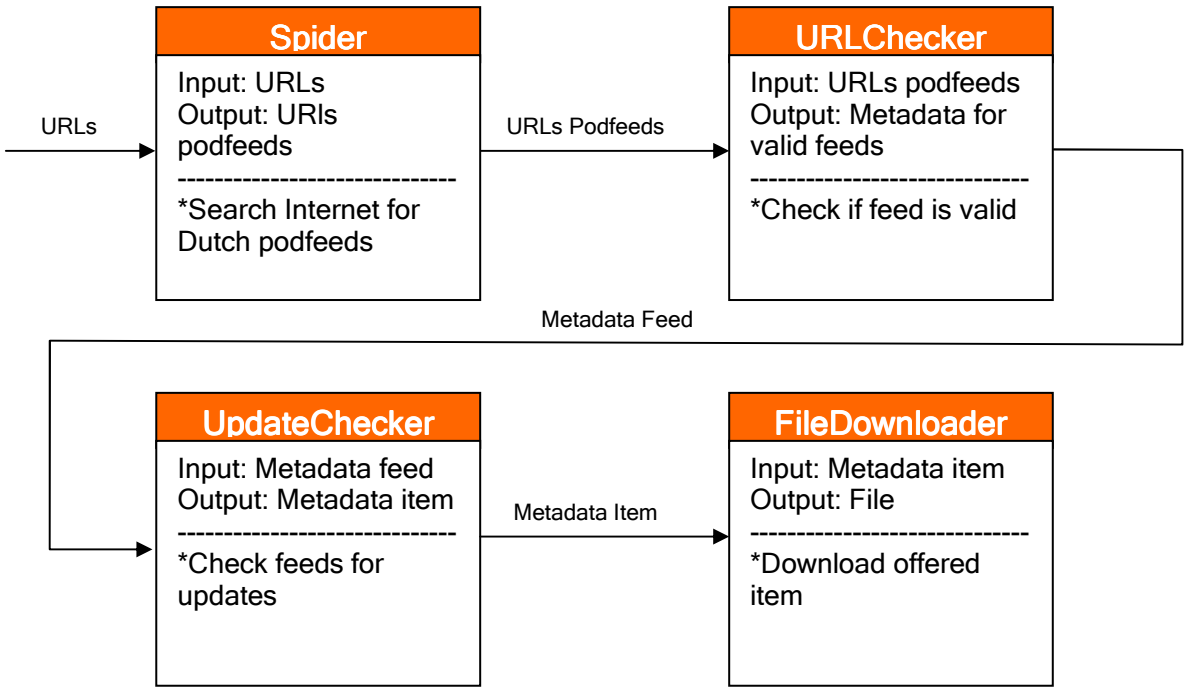


Figure 11.2: Acquire System Architecture

PodVinder			
Directory	*Sub-directory content*	*Sub-sub-directory content*	*Explanation*
feeds	$feed_1$.xml...$feed_n$.xml		Directory with feed XML-files: created by urlchecker.pl
files	$feed_1$...$feed_n$	$feed_{1.1}$.mp3...$feed_{1.t}$.mp3 ..... $feed_{n.1}$.mp3...$feed_{.n.t2}$.mp3	Directory with podcast-files, downloaded by filedownloader.pl
items	$feed_1$...$feed_n$	$feed_{1.1}$.xml...$feed_{1.t}$.xml ..... $feed_{n.1}$.xml...$feed_{.n.t2}$.xml	Directory with podcast XML-files, created by updatechecker.pl
mirror	$feed_1$.xml...$feed_n$.xml		Directory with podfeeds XML-files downloaded by updatechecker.pl
scripts	spider.pl urlchecker.pl updatechecker.pl		

	filedownloader.pl		
urls	$feed_1$.txt...$feed_n$.txt		Directory with last urls found in feed

Table 11.1: Acquire Directory Architecture

11.2.1 Spider

Functional requirement(s): 1
Summary: Searches the Internet for valid podcastfeeds. Takes a list of urls as input. While visiting these sites it gathers other URLs offered on this site. Keeps a list of valid podcastfeeds.

Design:

```
1: READ list of URLs from txt-file
2: FOR each URL in list
3:    DOWNLOAD content
4:    IF podcastfeed
5:          SAVE podcastfeed
6:    ELSE
7:          SEARCH new URLs
8:          IF new URL
9:                ADD URL to list
10:         END IF
11:   END IF
12: END FOR
```

11.2.2 URLChecker

Functional requirement(s): 2-6
Summary: Checks if the feed is valid: language-tag with Dutch language code, title-tag, link-tag, description-tag and at least one enclosure-tag providing a direct link within an item-tag. It is also checks if all the items offered are in mp3-format to make sure the system is able to handle the podcasts. If the feed is classified as valid an XML-file with the following structure will be generated:

```
<feed>
      <url>LINK OF FEED</url>
      <title>USER-GENERATED TITLE FOR FEED</title>
      <link>USER-GENERATED LINK FOR FEED</link>
      <description>USER-GENERATED DESCRIPTION FOR FEED</description>
</feed>
```

Design:

```
1: READ list of URLs from txt-file
2: FOR each URL in list
3:    DOWLOAD feed
4:    IF download successful
5:          CHECK feed for language-tag with Dutch language code
6:          CHECK feed for title-tag
7:          CHECK feed for link-tag
8:          CHECK feed for description-tag
9:          CHECK feed has at least one enclosure-tag with mp3-file
10:         CHECK feed has no other file-format in enclosure-tags
11:   END IF
14:   IF feed valid
15:         CHECK for duplicate
16:   END IF
17:   IF feed valid & not duplicate
```

```
18:         WRITE XML feed
19:   END IF
20: END FOR
```

11.2.3 UpdateChecker

Functional requirement(s): 7
Summary: Checks each feed in the system for updates. For each new item that is found an XML-file with the following structure will be generated.

```
<podcast>
      <feed>URL OF FEED THAT PROVIDED PODCAST</feed>
      <item>USER-GENERATED METADATA FROM FEED</item>
      <info>HEADER-INFORMATION</info>
      <id3v1>ID3v1-TAG INFORMATION</id3v1>
      <id3v2>ID3v1-TAG INFORMATION</id3v2>
      <classification>PODCAST CLASSIFICATION</classification>
      <asr>TEXT GENERATED BY ASR</asr>
</podcast>
```

Because the podcast is not yet classified, checked for header-information, ID3v1-tag, ID3v2-tag and put through the automatic speech recognition (ASR) the body of these tags will be left empty.

Design:

```
1: READ directory with feed XML-files
2: FOR each file in directory
3:    DOWNLOAD feed
4:    IF server status code not 304 && download successful
5:          READ last found urls-file
6:          FOR each item in downloaded feed
7:                CHECK valid item
8:                IF valid item
9:                      IF equal URL item, URL in last found-urls file
10:                           RETURN
11:                     ELSE
12:                           WRITE XML item
13:                     END IF
14:               END IF
15:         END FOR
16:         UPDATE last found urls-file
17:   END IF
18: END FOR
```

11.2.4 FileDownloader

Functional requirement(s): 8
Summary: For each podcast XML-file the item mentioned in the enclosure-tag will be downloaded. If the download fails the XML-file will be removed.

Design:

```
1: READ directory with podcast XML-files
2: FOR each file in directory
3:    DOWNLOAD podcast
4:    IF download successful
5:          SAVE podcast
6:    ELSE
7:          DELETE podcast XML-file
8: END FOR
```

12 Appendix E – Technical Documentation Analysis

The second part of the system has several smaller tasks that have to be performed. First of all the ID3-tags and file information has to be extracted from the file. After that all the metadata about the podcast is collected is bundled and used to classify the podcast as music or speech. Depending on the classification a transcription is generated by the speech recogniser. All this information is then added to the metadata of the podcast. All these steps can be summarised in one basic task:

- Analyse podcast to gather more information.

Taking the basic tasks and scope of the project into account, the following issues have been considered:

Issue 1: What programming language will be used?
Decision: The decision was made to use Perl taken the smaller tasks into account. Perl offers an easy package to extract information from mp3s. Also the file handling and text processing with regular expressions that has to be done a lot is fairly easy.

Issue 2: Which system will be used for automatic speech recognition?
Decision: For automatic speech recognition the UT-BN2002 system will be used. This system is specialised in broadcast news and is the subject of ongoing research.

12.1 Requirements

The requirements are based on the basic tasks and issues discussed. Also the scope of the project, answering a part of the research question and to serve as a foundation for further research, is taken into account.

12.1.1 Functional Requirements

1. Check for header-information
2. Check for ID3v1-tag
3. Check for ID3v2-tag
4. Classify item speech/music
5. Convert audio to text

12.1.2 Non-Functional Requirements

1. Modular

12.2 Implementation

To create flexibility the system architecture consists of modules (see Figure 12.1) that each has their own part in performing the basic tasks.

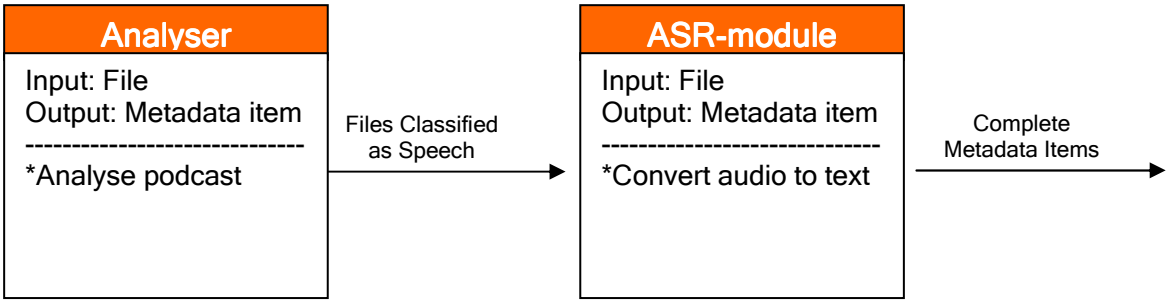


Figure 12.1: Analyse System Architecture

The output is saved in a directory structure to create easy access for modules or even other applications build to use the generated information. The output-handling and directory structure also makes it possible to easily monitor and analyse the system (see Table 12.1 for details).

PodVinder			
Directory	*Sub-directory content*	*Sub-sub-directory content*	*Explanation*
asr	ASR-module		
hyp	$feed_1$.xml...$feed_n$.xml	$feed_{1.1}$.mp3...$feed_{1.t}$.mp3 ..... $feed_{n.1}$.mp3...$feed_{n.t2}$.mp3	Directory with hypothesis text: created by ASR-module
index	$feed_1$.xml...$feed_n$.xml	$feed_{1.1}$.mp3...$feed_{1.t}$.mp3 ..... $feed_{n.1}$.mp3...$feed_{n.t2}$.mp3	Directory with complete podcast XML-files, ready to be indexed: created by ASR-module/analyser.pl
scripts	analyser.pl combine.pl		

Table 12.1: Analyse Directory Architecture

12.2.1 Analyser

Functional requirement(s): 1-4
Summary: Each mp3-file downloaded in the files-directory is analysed on several aspects. The header-information, ID3v1-tag, ID3v2-tag is retrieved from the file. After this information is retrieved the podcast is classified as speech/music by analysing the metadata gathered. The classifier, discussed mentioned in paragraph 4.4, used has been trained with 700 items. The XML-file is updated with this information.

Design:

```
1: READ directory with podcasts
2: FOR each file in directory
3:    EXTRACT header-information
4:    IF ID3v1 exists
5:          EXTRACT ID3v1-tag
6:    END IF
7:    IF ID3v2 exists
8:          EXTRACT ID3v2-tag
9:    END IF
```

```
10:   CLASSIFY speech/non-speech
11:   WRITE XML
12: END FOR
```

12.2.2 ASR-module

Functional requirement(s): 5

Summary: Text of the audio is generated when the podcast is classified as speech. First of all the podcast is converted from mp3 to raw audio. Secondly the UT-BN2002 system is used to generate a hypothesis text. The system is a hybrid RNN/HMM system with a 65K vocabulary and a statistical trigram language model trained on a newspaper corpus that has a word-error-rate of about 30% on broadcast news shows. At last the hypothesis is added in the XML-file (combine.pl) of the podcast and moved to the index directory.

Design:

```
1: READ directory with podcasts
2: FOR each file in directory
3:    IF classification is speech
4:          CONVERT mp3 to raw audio-file
5:          RUN UT-BN2002 on file
6:          WRITE hypothesis text
7:          UPDATE XML
8:    END IF
9:    MOVE XML
10: END FOR
```

13 Appendix F – Technical Documentation Search

The third part of the system has to perform two basic tasks. First all the metadata that has been collected and generated in the first two parts of the system have to be put into an index. Second the system has to be able to retrieve results from this index based on a query given by a user. To summarise:

- Index generated metadata of podcasts.
- Convert queries from users to results from index.

Taking the basic tasks and scope of the project into account, the following issues have been considered

Issue 1: Which system will be used for the search engine?
Decision: Lucene will be used as search engine. The Lucene-project is open source search engines and can be easily adapted for own usage.

Issue 2: How are documents scored?
The standard scoring class of Lucene will be used. Lucene scoring uses a combination of the Vector Space Model (VSM) of information retrieval and the Boolean model to determine how relevant a given document is to a user's query[1].

13.1 Requirements

The requirements are based on the two basic tasks and issues discussed. Also the scope of the project, answering a part of the research question and to serve as a foundation for further research, is taken into account:

13.1.1 Functional Requirements

1. Index podcast metadata
2. Convert input from user to query for database
3. Return results based on query

13.1.2 Non-Functional Requirements

1. Modular

13.2 Implementation

The system architecture (see Figure 13.1) is based on basic classes that are offered by Lucene.

[1] Apache Lucene – Scoring, http://lucene.apache.org/java/docs/scoring.html

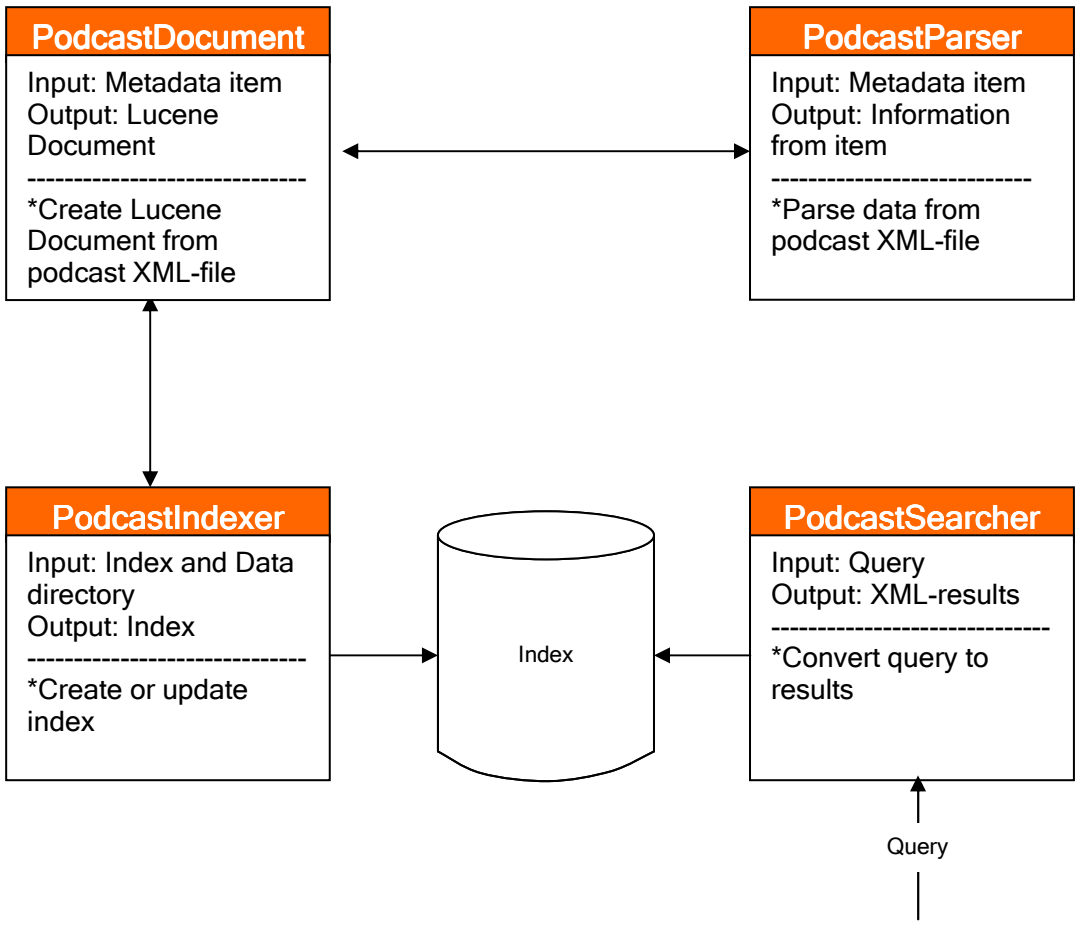


Figure 13.1: Search System Architecture

13.2.1 PodcastDocument

Functional Requirement(s): 1-3
Summary: A utility for making Lucene Documents from podcast XML-files. Based on the standard Lucene Document class.

Design:

```
public static Document Document( File f )
1: CREATE new Document
2: CREATE new PodcastParser
3: ADD information to Document using PodcastParser
4: RETURN Document
```

13.2.2 PodcastParser

Functional Requirement(s): 1-2
Summary: A utility to parse information from podcast XML-files.

Design:

```
public PodcastParser( File file )
1: CREATE new FileReader
2: READ podcast XML-file

public String getFeed()
1: RETURN removeTags( feed from podcast )
```

```
public String getItem( String tag )
1: IF tag is "total"
2:    RETURN removeTags( full metadata item )
3: ELSE if tag is <tag>
4:    RETURN removeTags( metadata item-<tag> )
5: END IF

public String getInfo( String tag )
1: IF tag is "total"
2:    RETURN removeTags( full info file )
3: ELSE if tag is <tag>
4:    RETURN removeTags( metadata info-<tag> )
5: END IF

public String getID3v1( String tag )
1: IF tag is "total"
2:    RETURN removeTags( full metadata ID3v1 )
3: ELSE if tag is <tag>
4:    RETURN removeTags( metadata ID3v1-<tag> )
5: END IF

public String getID3v2( String tag )
1: IF tag is "total"
2:    RETURN removeTags( full metadata ID3v2 )
3: ELSE if tag is <tag>
4:    RETURN removeTags( metadata ID3v2-<tag> )
5: END IF

public String getClassification()
1: RETURN removeTags( classification )

public String getASR()
1: RETURN removeTags( ASR )

private String removeTags( String line, String tag )
1: RETURN line with <tag> removed
```

13.2.3 PodcastIndexer

Functional Requirement(s): 1-2
Summary: Creates a new index from a directory with podcast XML-files or adds new podcast XML-files to existing index. Based on the standard Lucene Indexer class. The standard IndexWriter from the Lucene package is used to write index.

Design:

```
public static void index( File indexDir, File dataDir )
1: CREATE IndexWriter
2: USE function indexDirectory

public static void indexDirectory( IndexWriter writer, File f )
1: FOR each item in f
2:    IF directory
3:          use indexDirectory
4:    ELSE IF
5:          use indexFile
6:    END IF
7: END FOR

public static void indexFile( IndexWriter writer, File f )
1: USE PodcastDocument.Document
```

```
2: INDEX Document
```

13.2.4 PodcastSearcher

Functional Requirement(s): 3
Summary: Searches the index with the given query, returns a XML-document if any documents are found. Based on the standard Lucene Searcher class. The XML-document has the following structure:

```
<results amount="#RESULTS">
      <result id="#ID" score="SCORE">
            <title>TITLE OF PODCAST</title>
            <download>DOWNLOAD LOCATION PODCAST</download>
            <feed>LINK OF FEED PODCAST WAS FOUND</feed>
            <description>DESCRIPTION OF PODCAST FOUND IN FEED</description>
            <duration>DURATION OF PODCAST in MM:SS</duration>
            <size>SIZE OF PODCAST IN MB</size>
      </result>
</results>
```

The standard Searcher, QueryParser, Query and Hits from the Lucene package are used to retrieve documents from the index.

Design:

```
public void search( String queryString, String indexDir )
1: CREATE new Searcher
2: CREATE new QueryParser
3: CREATE new Query
4: SEARCH index with Query to get Hits
5: FOR each item in Hits
6:    CREATE XML results
7: END FOR
8: WRITE XML results
```

14 Appendix G – Technical Documentation Presentation

The last part of the system provides the interface for interaction between the user and the system. Users have to be able to give the system a simple query or a more advance queries taking certain preference of the user such as duration and size of the desired podcast into account. Once the query is entered using the interface, the query is given to the searcher, which returns results in XML. The user interface has to parse these results and present them in an orderly fashion. To summarize, the last part of the system has to perform two basic tasks:

- Retrieve and send user query to searcher.
- Present results.

Taking the basic tasks and scope of the project into account, the following issues have been considered.

Issue 1: Which language will be used?
Decision: The interface will be build using OpenLaszlo. OpenLaszlo is an xml-based mark-up language with javascript abilities that can be converted to flash or dhtml-files for publishing. The decision for OpenLaslzo is based on the XML capabilities of the language, making it very easy to base the interface on XML-files that are received. A java server applet will be used based on the PodcastSearcher class used in part three of the system to be able to search through the index.

Issue 2: On which criteria can the user search?
Decision: The user will have a basic and advance search option. The basic option just offers the user the possibility to input a query with results returned based on user-generated metadata and transcripts generated by the ASR if available. The advance search option will make it possible to combine the query with a category (speech/music/speech & music), a duration and maximum size. Also the option is offered to search only through user-generated metadata or transcripts produced by ASR.

Issue 3: Which information of the results is displayed?
Decision: The most interesting of the podcast is displayed for to the user: title of podcast as link to the podcast, feed that contained podcast, excerpt where a result is based on, duration and size.

Issue 4: How many results are displayed on one page?
Decision: Only five podcast will be shown at once to limit the amount of information given to the user. The possibility is offered to scroll through all the results returned by the system.

14.1 Requirements

The requirements are based on the two basic tasks and issues discussed. Also the scope of the project, namely answering a part of the research question and serving as a foundation for further research, is taken into account:

14.1.1 Functional Requirements

1. Users can perform a general search for podcasts based on a query.
2. Users can perform an advance search for podcasts based on a query, category (music / speech / music & speech).
3. GUI presents five retrieval results at a time.
4. Users can scroll through the retrieved podcasts.

14.1.2 Non-Functional Requirements

1. Simple interface.

14.2 Implementation

The system architecture (see Figure 14.1) is divided into four parts: the GUI (Graphical User Interface), the searcher (results.jsp), library classes (lucene.jar) for the searcher and the index. The index is the only part which is not located on the web server.

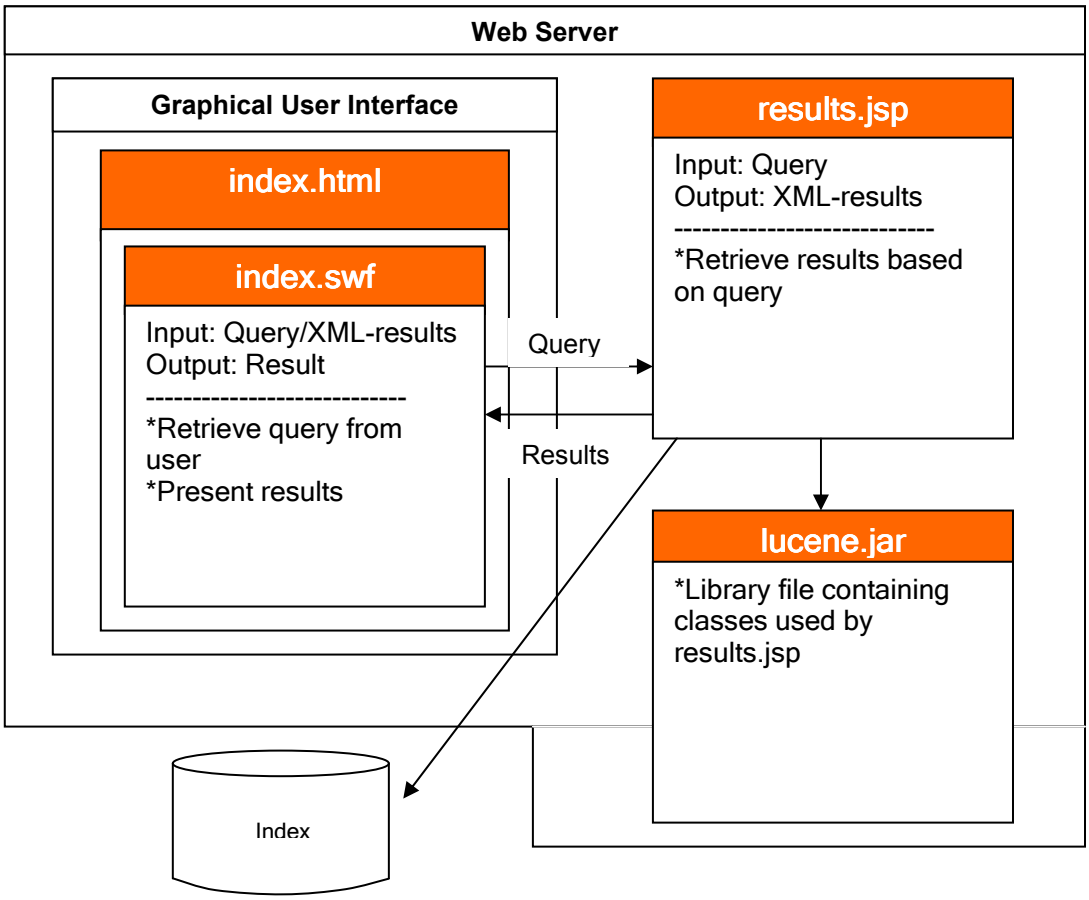


Figure 14.1: Presentation System Architecture

14.2.1 GUI

Functional Requirement(s): 1-4
Summary: The graphical user interface gives the user the ability to give the system a query. It is also responsible for presenting the results to the user. The GUI consists of a shell that contains the flash-file. Queries are posted to results.jsp to retrieve results from the index.

Design:

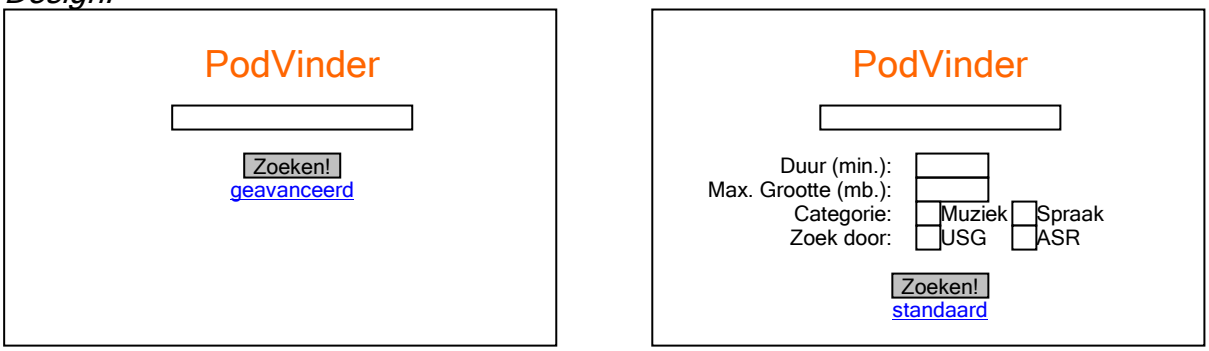


Figure 14.2: Design of the basic (left) and advance (right) search interface.

Figure 14.3: Design of result interface

14.2.2 Results.jsp

Functional Requirement(s): -
Summary: results.jsp is the PodcastSearcher discussed in paragraph 13.2.4.. The file is converted from a java class into a Java Server Applet (JSP) so it can be used in combination with the GUI and the webserver. The applet is able to retrieve POST variables that are posted from the GUI and use them to perform a search in the index.
Design: See paragraph 13.2.4.

14.2.3 Lucene.jar

Functional Requirement(s): -
Summary: lucene.jar is a library class containing the necessary classes from Lucene to make results.jsp work.
Design: See http://lucene.apache.org/java/2_2_0/api/index.html