Computer Science Department
University of Kaiserslautern, Germany

Multimedia Analysis and Data Mining Competence Center
German Research Center for Artificial Intelligence (DFKI GmbH)
Kaiserslautern, Germany

Combining Social and Content Based Signals for Personalized Tag Suggestion on YouTube

Bachelor Thesis

Author:	Dominik Henter
Supervisor:	Dr. Adrian Ulges
	Damian Borth, M. Sc.
Reviewer:	Prof. Dr. Andreas Dengel
	Dr. Adrian Ulges
Submission Date:	30 May, 2012

I declare that this document has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. It has not been accepted in any previous application for a degree. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Dominik Henter
30 May, 2012

From a bit to a few hundred megabytes, from a microsecond to half an hour of computing confronts us with completely baffling ratio of 10^9! The programmer is in the unique position that his is the only discipline and profession in which such a gigantic ratio, which totally baffles our imagination, has to be bridged by a single technology.

E. W. Dijkstra

Abstract

Tag suggestion can considerably enhance a user's experience on a social sharing website by providing reasonable proposals of possible tags, which the user can easily use for his/her content. With this the tags' quantity and quality is effectively increased, as the user needs less time to provide more tags and may use tags that are highly fitting but that he/she would not have thought of him-/herself.

While most current tag suggestion systems in use today rely on tags the user used in his previous videos, sometimes enhanced by tags that occur together with these tags in videos uploaded by other users, this thesis describes tag suggestion systems that incorporate several modalities. The first contribution is a comparative study on several systems that rely on different modalities for tag suggestion on YouTube, like the user's tag history, the user's activity on the social sharing platform (social signals) and the visual information of the uploaded content (content based signals). The second contribution is the Visual Personalized Tag Transfer system that combines both personal information gained from the user's history and information gained from videos that are visually similar to the uploaded one, to outperform a purely history based system, which can be considered the standard approach. The third contribution are two approaches to "fuse" multiple systems into a single multimodal tag suggestion system are described, with the Weighted Sum based fusion showing great potential.

Zusammenfassung

Tag Suggestion kann die Benutzerfreundlichkeit einer Social Sharing Webseite beachtlich verbessern, indem sie sinnvolle Tags vorschlägt, die der/die Benutzer/in leicht für seine/ihre eigenen Inhalte nutzen kann. Dadurch kann sie effektiv die Tag-Quantität und Qualität steigern, da der/die Benutzer/in weniger Zeit braucht, um mehr Tags vorzuschlagen, und da er/sie möglicherweise gut passende Tags finden kann, auf die er/sie selbst nicht gekommen wäre.

Während die meisten aktuellen Tag Suggestion Systeme auf Tags aufbauen, die der/die Benutzer/in für vorherige Videos benutzt hat – manchmal durch Tags, die zusammen mit diesen Tags in den Videos anderer Benutzer vorkommen, erweitert – beschreibt diese Arbeit Tag Suggestion Systeme, die mehre Modalitäten nutzen. Der erste Beitrag ist eine Vergleichsstudie diverser Systeme, die verschiedene Modalitäten nutzen, um Tags auf YouTube vorzuschlagen. Beispiele hierfür sind die Tag-Historie, die Aktivität des/der Benutzers/Benutzerin auf der sozialen Plattform (sog. social signals) und die visuellen Informationen der hochgeladenen Inhalte (sog. content based signals). Der zweite Beitrag ist das Visual Personalized Tag Transfer System, welches sowohl die Tag-Historie des/der Benutzers/Benutzerin als auch Informationen, die durch visuell ähnliche Videos gewonnen werden können, nutzt, um ein ausschließlich auf der Historie basierende System zu übertreffen, welches als Standardansatz betrachtet werden kann. Der dritte Beitrag sind zwei Ansätze zum Kombinieren mehrerer Systeme in ein einzelnes multimodales Tag-Suggestion-System. Hierbei zeigt die Weighted Sum basierte Kombination besonders großes Potential.

Contents

1. Introduction

Video on the Internet has become more and more important over the last several years and has become a crucial part of the Internet and the everyday life of millions of people. Private and professional videos alike are uploaded and shared with millions of others. This is shown by YouTube, which is the most visited video sharing platform in the Internet (YouTube is ranked third by Alexa, only outranked by Google and Facebook [1]).

The most viewed video on YouTube was viewed nearly 730 million times ([28]) and over four billion videos are viewed every day, showing how widespread the activity of watching videos is. In only one second a whole hour's worth of video is uploaded to YouTube, illustrating that creating and sharing self created content is likewise widespread. And the social awareness of YouTube's users is shown by the fact that 100 million people take one of the available social actions, like commenting other videos or sharing videos with friends, in one week ([29]).

To handle such vast amounts of video a well working search engine is mandatory. As videos do not come with associated text by themselves, like websites do, and as the title alone is often not descriptive enough to be a reliable source of information for a search engine, users have the possibility to associate their videos with a list of additional words, so called "tags". An example of a video and its associated tags on YouTube can be seen in Figure 1.1. Tags do not only help the search engine in retrieving more suitable videos, but they also allow to find videos that share the same tags. Furthermore, a video's tags might be the basis for recommending other potentially interesting videos while watching it. All this works better the more tags the users use for their videos and the more descriptive these tags are.

To aid the user in finding reasonable tags or in tagging at all, thus increasing both quality and quantity of tags (compare [2]), many sites utilize so called tag suggestion systems. They provide a set of tags which the user can easily choose from (usually by simply clicking on them or by providing them as autocompletion options) to annotate his content alongside his own tags. The user interface of the tag suggestion system as implemented in YouTube can be seen in Figure 1.2.

The tag suggestion technology can also be used to deduce tags for videos that are on-line but were not tagged by their original uploader. This again can be helpful for text-based search engines. Additionally this can be used for targeted advertising in popular but untagged videos. This means that advertisements can be shown that suit the video's contents, based on the automatically suggested tags. As tag suggestion systems normally do not only provide an unordered list of suitable tags but also probability scores for each tag, this can be used to even further optimize text-based search engines that, without

Figure 1.1.: A video as seen on YouTube. Associated with this video are so called "tags" that describe its content and enable other users to find it. Highlighted with a red frame are the tags as visualized on YouTube.

such scores, have to treat all tags given to a video the same, even though the tags might not be equally important.

It is still often stated that tag suggestion systems that use the tags that occur most often in the user's own tagging history (i.e. the tags he used for previous uploads) are "a reasonable upper bound"[21] for the performance of such a system. In contrast to this, this thesis describes systems that use multiple modalities, which has shown promising in recent research, as seen in [18, 27]. A modality, in this context, means a source of information, like the most common tags on a platform. The modalities used in this thesis are:

- The user's own tagging history, an approach that, as said before, performed well in the past and is a good way to find personalized tags. For example, a user who only used cat related tags like `cat` (3 times), `yarn ball` (3 times) and `cat toy` (2 times) for his three previously uploaded videos will be suggested exactly those tags (ordered by the number of times they were used) for a newly uploaded video, independent of its contents.
- The tag co-occurrence approach that utilizes tags co-occurring with the user's history for tag suggestion. This approach is used to have an explorative element in contrast to the history based system, as it has the ability to suggest tags that the user did not use before. For example, other users that upload cat related videos might use more specific tags like `cat dancing`, `tomcat` or `kitten` that might as well describe the example user's cat and therefore might be suitable tags for his videos.

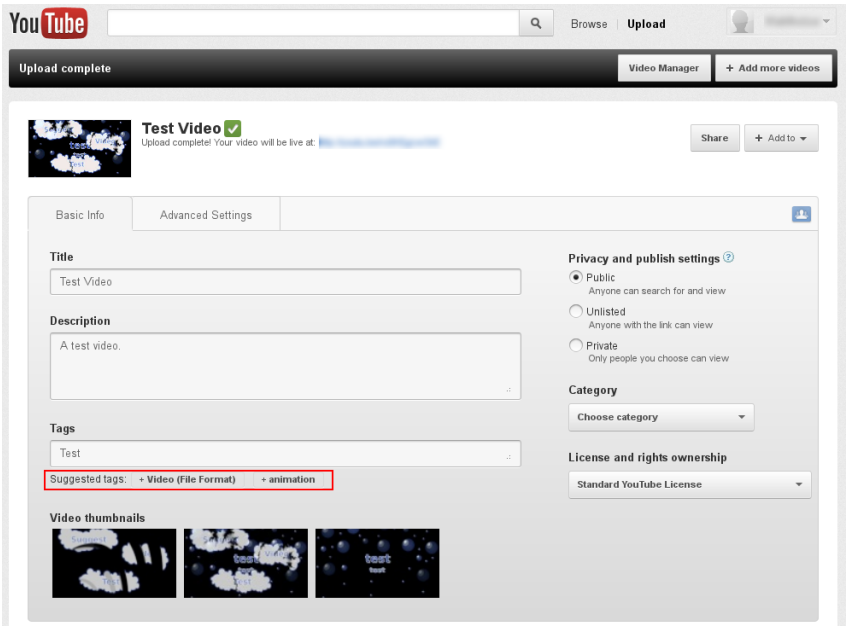


Figure 1.2.: The tag suggestion interface as it is implemented in YouTube. Clicking on one of the suggested tags (highlighted by a red frame) adds it to the video.

- An approach using exemplary social signals gained from the user's subscribed channels. This is a way of harnessing the user's social behavior for tag suggestion, which potentially reflects the user's interests and with this might yield suitable tags. For example, the same user might have subscribed to cat related channels, in which more elaborate tags like `feline` might be used that then would be suggested to him/her.
- A system based on visual signals (namely a Visual Words based approach using SIFT features, which is explained in detail in Section 3.6.2) gained from the uploaded content itself. It is capable of finding relations between similar (in terms of visual similarity) videos that do not have any text based or social background based relation, potentially finding suitable videos and their tags that other systems could not. For example, the newly uploaded video of the example user might not show cats but scenes of his/her daughter's birthday. Videos that are visually similar might provide tags like `birthday` or `cake`.

In addition a system is proposed that is not based on one modality but on two. It merges the user's previously uploaded videos, which are re-ranked by visual information, with visually similar videos not uploaded by the user, using a globally chosen weighting factor, determined by grid search[1]. This system might be able to unite the reliability of the history modality with the adaptability of the visual signals. For example if the example user uploads an outdoor cat video in which the cat is not prominent, the history will still provide cat based tags, whereas the visual signals might suggest tags related to gardening. On the other hand if he/she uploads a video of his/her daughter's first soccer

[1]See Section 4.5.1 of [25] for details.

match, the history's cat related tags are not suitable but the visual signals might suggest soccer related tags.

This thesis' main goal is to create a working tag suggestion system that surpasses the performance of current systems in a real life setup and, while doing so, to provide insights regarding the usefulness of each system and the combinations thereof with a realistic evaluation. For this not only the systems as described above are proposed, but also two approaches to the fusion of those systems are presented and evaluated, as well as a way of finding the correct parameters for these fusions:

- A simple rule based system that is based on knowledge gained during research. It uses static rules to pick one of the results provided by multiple tag suggestion systems based on a user's specifics.
- An approach that uses the weighted sum[2] of either scores or ranks (similar to the Borda count method[3]) of the tags provided by four of the systems described above. Here, several approaches for choosing the correct weights are taken. The first is to search a single global weight combination via grid search that works for every user. The second approach is to find individual weight combinations for every video, again using grid search – this is considered an oracle approach, as the performance value the grid search is optimized on cannot be considered known when a video is freshly uploaded. Lastly, individual weights are learned from similar users (in terms of features that are described in more detail in Section 3.8.2) in a leave-one-out fashion[4].

All of this is done not only with the goal of surpassing the History based system, but also to provide insights regarding the utility of the different approaches.

Challenges

One of the main challenges for the proposed systems is the high degree of personalization of the suggested tags that they aim at. Many papers in this field use user studies to evaluate their systems, in which several people decide whether a given tag fits the content or not. This results in findings like an "average relevance"[26] of 43% when evaluating tags given by the original uploader and is easily deceived by predominantly using general tags (compare [26]) – "music" fits every music video, whereas "rock" or even "Rolling Stones" is suitable for a considerably smaller subgroup only. The aim of this thesis is to try to predict the tags the user actually used, regarding this as a meaningful measurement for how far the tag suggestion system is personalized and how far it is able to suggest tags a user would actually use him-/herself. To verify this the systems are evaluated on the tags originally given by the provider of the content.

Another challenge is the high diversity inherent in such openly available social sharing websites as YouTube and the sparsity of data that might come with this. Every user

[2]See Chapter 2 of [17] for general calculus or Section 6.2 of [5] for the weighted sum model in decision making.

[3]See [4] for the original conception as conceived by Jean Charles Borda.

[4]See Chapter 24 of [7] for an overview.

has different interests and a different count of past videos and subscribed channels, one user is actively uploading content all the time, another user just watches videos. To cope with this, several modalities are used to have enough information to still be able to discriminate users in every case and, furthermore, several features describing users are investigated in terms of their usefulness to find similar users in spite of this highly diverse environment. With the diversity of interests also comes visually diverse data. To catch the visual diversity, the concept detection based approaches are trained on a high number of diverse concepts (a full list can be found in Table A.1).

Outline

The rest of this thesis is organized as follows: Chapter 2 introduces related work that dealt with similar problems, or parts thereof, in the past and will discuss the need of novelty in the field of tag suggestion and in how this thesis seeks to help satisfy this need. In Chapter 3 five groups of tag suggestion systems are presented, each using different modalities for ranking and providing tags, as they were listed above. This is followed by Chapter 4, where the presented systems and their fusions are evaluated qualitatively in a real life setting, with a special focus on the performance when using real-world tags as ground truth as they were generated by the user. The thesis is then concluded by Chapter 5, where the findings of Chapter 4 are briefly summarized and interpreted. In addition an outlook on possible future improvements is given.

2. Related Work

As tagging is still a popular way of retrieving content and making it findable for others, and as tag suggestion systems have been accepted as possible means to increase tag quality and quantity, a lot of tag suggestion systems have been proposed in the past.

In [24] TagAssist, a tag suggestion system for blogs, is introduced that infers tags for a new blog post based on similar – in terms of TFIDF[1] – posts from a corpus of known posts. The results are refined with respect to the popularity of the blog, where the post originates from, shared topics, frequency of the tags and tag count in the corpus. This system neglects potential benefits from personalization and makes only limited use of the social structure provided (i.e. the blog popularity), furthermore it is hardly applicable to problems outside of the text domain.

Some of these shortcomings are addressed in [23] where existing tags, provided by a different tag suggestion system or the user, are enhanced by means of tag co-occurrence in a vocabulary provided by tagged images taken from Flickr. This allows to use this system in virtually every content domain. And [13], in which, based on concept vocabularies and an image's visual information (i.e. color and texture features), tags are suggested for an untagged image, as well as in [26], where the audiovisual information provided by a newly uploaded video is used to find appropriate tags for it with the help of AdaBoost[2], both allowing tag suggestion without user knowledge or a social structure for images and videos respectively. In contrast to this thesis, which proposes the Visual Personalized Tag Transfer approach that combines user based and content based signals to gain the benefits of both personalization and content information, all three systems make no or non-exhaustive use of personalization and rely on only one modality.

Problems that other approaches seek to rectify with higher personalization. For example via hierarchical clustering while taking user interest into account as [22]. Or by using multimodal approaches like [27] that use content and tag correlation together with co-occurrence. Or [18] that personalizes, is multimodal and, in contrast to [22], even considers the social structure, by utilizing the user's personal context (i.e. the tags used in the past), contacts, groups and the collective context (i.e. the tags from all resources) but lacks content based signals. Another approach, described in [21], combines the visual information, using ALIPR[3], and the user's social background (called "Local Interaction Networks") but ignores the user's own tagging history under the assumption that this information might not be available. The evaluation in [21] shows that the user's own tagging history outperforms their proposed system, indicating that this modality should

[1] Described in [20].
[2] See [10] for an explanation of AdaBoost.
[3] See [13] for information about ALIPR.

be used whenever possible. [14] seeks to surpass the history based system by subsequently personalizing the tags given by the user's own history using cross-entropy[4] and succeed, albeit neglecting the content's visual information. To address these problems, this thesis proposes the Weighted Sum based fusion, a system that combines multiple modalities and is easily extensible. For this a way of learning the appropriate weights on a per user basis is proposed.

This thesis also provides an evaluation of the single systems for each modality, as well as an evaluation of their fusions. Therefore serving as a comparative study on tag suggestion systems. The evaluation is done on real life conditions without the heavy restrictions made in several of the papers above, i.e. the dataset is only filtered by a list of Stop Words and the suggested tags are evaluated on the original uploader's tags rather than using user studies.

[4]For information on the cross-entropy algorithm see [19].

3. Tag Suggestion Systems

In this chapter several tag suggestion systems and their fusions – the combination of the single system's (intermediary) results – are described. For each a motivation is given. For this the general notion of a tag suggestion systems is clarified. Furthermore, the tag suggestion systems that are used for the fusions are shown in detail, as well as two additional systems that are not used in any of the fusion approaches, but do provide additional insights. These systems are divided in sections based on the modalities they rely on for suggesting tags. This chapter ends with proposing several ways of fusing the single systems into one, again subdivided into the single approaches.

The systems and their fusions will be described in a manner that tries to be as general as is possible without becoming unnecessarily complex. This means that the underlying social platform is not assumed to be YouTube in specific, but a general social platform that has to fulfill as few requirements as possible, which will be described in Section 3.1. Additional requirements for specific systems will be noted in the sections of the respective systems that require them. Even if some requirements are stated, the way of description tries to be easily adjustable to other platforms and resources that at least partly fulfill these requirements (e.g. if the visual content of videos is used for a system, the information gained from the visual content of images can be used with nearly the same algorithm).

3.1. Tag Suggestion Systems in General

The set of all videos that have been uploaded on the platform is denoted as $\mathcal{V}$ and additional information about the videos $v \in \mathcal{V}$ and their associated users u_v is considered known, including, but not limited to, the corresponding tags of the video T_v and the videos previously uploaded by the user u_v, denoted as H_{u_v}, a potentially empty set of videos. The set of all tags used in at least one video in $\mathcal{V}$ is denoted as $\mathcal{T}_\mathcal{V}$, representing the global vocabulary. The general setup of a tag suggestion system is illustrated in Figure 3.1. A tag suggestion system takes an untagged video v_{new} uploaded by a user $u_{v_{new}}$ and suggests a list of tags $T_{v_{new}} \subseteq \mathcal{T}_\mathcal{V}$ that is ordered by likelihood (the first element being the most likely). Each tag $t \in T_{v_{new}}$ is associated with its respective likelihood score, denoted as $score_t$.

In a real life scenario the list of suggested tags $T_{v_{new}}$ would be capped after a reasonable amount of tags (e.g. 25) as to not overflood the user with too many, potentially inaccurate, tags. The kind of tag suggestion system considered in this thesis works without any user interaction and has no prior knowledge of the considered video but may make use of prior knowledge about the user if such knowledge is available.

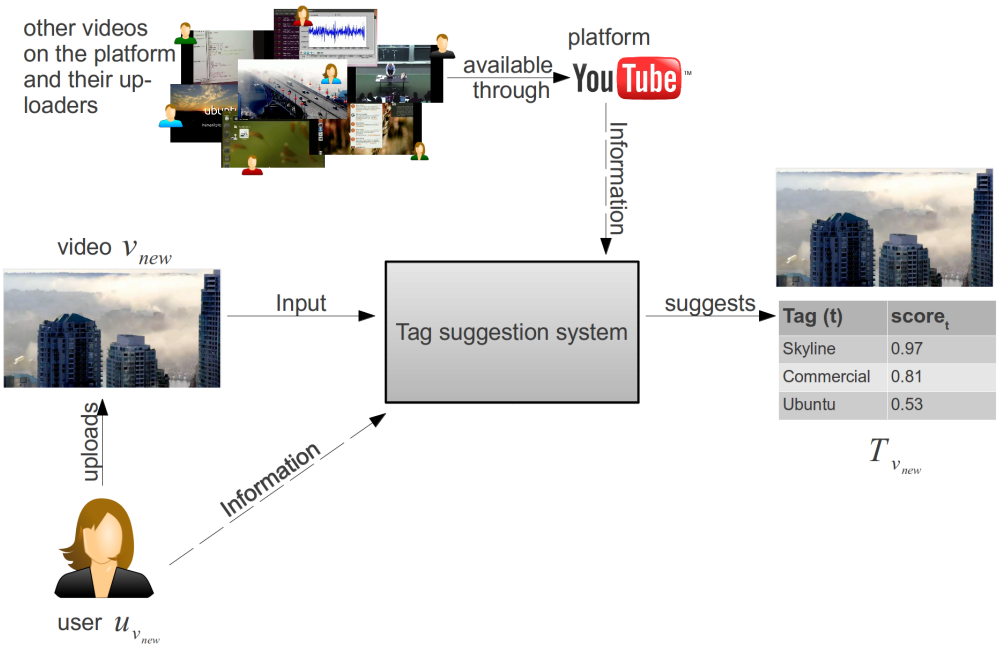


Figure 3.1.: The general setup of a tag suggestion system.

3.2. Baseline: Global Tag Statistic Based System

Motivation

The Global Tag Statistic based system is designed to be simple and easily computable, but also to provide tags in every circumstance (an ability not all of the following systems will posses). It illustrates the capabilities of a system that uses neither information about the user nor of the uploaded content. Furthermore, it provides a reasonable baseline for other tag suggestion systems, a system performing worse than this should not be considered for fusion, as it would most likely only produce noise.

Description

This system uses the global statistic created over the tags of all available videos to determine the overall most used tags of a platform. The ordered list of these global tags is then used as the suggestion for all considered videos. For this, the tags of all users, or a reasonable subset of users, must be known, a knowledge that is provided by the underlying social platform. Algorithm 1 illustrates how such a global tag statistic can be created.

The set $\mathcal{T}_\mathcal{V}$ (the set of all tags of all videos in V) is ordered by the occurrence of each tag occ_t (the most frequent one being at the top) and each tag is associated with its normalized score $score_t = \frac{occ_t}{|\mathcal{T}_\mathcal{V}|}$, as a measure for the accuracy of the suggested tag. This normalization is motivated by the fact that $|\mathcal{T}_\mathcal{V}|$ is the maximum number of times a tag

Algorithm 1 Create global tag statistic

```
for all v ∈ 𝒱 do
    for all t ∈ T_v do
        occ_t += 1                          ▷ occ_t is initialized with 0 for all t
    end for
end for
```

could possibly occur, in the case that it would be the only tag used up till now. This would result in the maximum confidence for this system and a score of 1.0. For each newly uploaded video v_{new} this system suggests the same list of tags $T_{v_{new}} = \mathcal{T}_{\mathcal{V}}$ independent of the user or the video's content.

3.3. History Based System

Motivation

As the History based system is still considered a standard approach for tag suggestion systems in general (compare [21]) and as this system is widely used on real platforms, this is a reasonable baseline to compare the systems proposed in this thesis with. Furthermore, this system can be seen as strongly personalizing, as it uses only tags that occurred in videos the user has uploaded before, a quality that might be of importance to reach the goal of suggesting exactly those tags that a user would actually use.

Description

The History based tag suggestion system utilizes a user's tagging history to deduce tags for a video recently uploaded by this user. To create the list of suggested tags $T_{v_{new}}$ for an uploaded video, the uploader $u_{v_{new}}$ is determined (as mentioned in the general section, this is considered to be information readily available by the underlying social sharing website) and the tagging history for this user is created as seen in Algorithm 2. The list

Algorithm 2 Construct tagging history for a specific user u

```
TH_u = ∅
for all v ∈ H_u do
    TH_u = TH_u ∪ T_v
    for all t ∈ T_v do
        histocc_t += 1                      ▷ histocc_t is initialized with 0 for all t
    end for
end for
```

of suggested tags is then $TH_{u_{v_{new}}}$ ordered by the number of occurrences in the history for each tag t $histocc_t$ and the score representing the system's confidence is computed as

$score_t = \frac{histocc_t}{\left|H_{u_{v_{new}}}\right|}$ for each tag. The normalization by $\left|H_{u_{v_{new}}}\right|$ is done, as each tag can occur only once in each video. The maximum normalized score of 1.0 is reached, if a tag occurs in every video. If the user $u_{v_{new}}$ has uploaded no videos before the upload of v_{new}, the list of suggested tags is empty, as no tagging history can be built.

3.4. Co-Occurrence Based System

Motivation

The motivation for this system is to include an exploratory component for tag suggestion: If the list of suggested tags always consists of the user's own vocabulary, there is no way of suggesting tags that the user has not used before, which might not reflect a user's actual tagging behavior. For example if a user is a winter sports enthusiast, he might have uploaded two videos about snowboarding, together with suitable tag like `snowboard, snow, X Games`. If the same user now uploads a video about his own attempts at skiing, the tags `snowboard` and `X Games` no longer fit, but other tags that occurred together with these two or `snow` might still be applicable. For example if a different user uploaded a video about his first skiing attempts tagged with `snow`, `first attempt`, `skiing` or a report about the last Winter X Games tagged `winter`, `X Games`, `snow`, `skiing`, `snowboarding`. Furthermore, the History based system might misleadingly encourage a user to reuse previous tags although they are not reflecting the actual content, simply because they are somewhat relevant and easier to use.

Description

For the Co-Occurrence based system, tags are collected that occur with the ones from the user's tagging history, e.g. if a video exists that is tagged with both `dog` and `cat`, these tags co-occur. The set of all tags co-occurring with a given tag t is denoted as CO_t and is calculated as depicted in Algorithm 3. To create $T_{v_{new}}$ $TH_{u_{v_{new}}}$ has to be calculated as

Algorithm 3 Construct set of co-occurring tags for a specific tag t

```
CO_t = ∅
for all v ∈ V do
    if t ∈ T_v then
        CO_t = CO_t ∪ T_v
        for all t_v ∈ T_v do
            cooc_{t→t_v} += 1        ▷ cooc_{t→t_v} is initialized with 0 for all pairs (t, t_v)
        end for
    end if
end for
```

described in Section 3.3. For a specific user u the set of all co-occurring tags COT_u for each tag in TH_u is calculated. This is shown in detail in Algorithm 4. In order to receive

Algorithm 4 Construct co-occurrence based set of tags for a specific user u

$COT_u = \varnothing$
for all $t \in TH_u$ **do**
  $COT_u = COT_u \cup CO_t$
  for all $t_{co} \in CO_t$ **do**
    $coococc_{t_{co}} \mathrel{+}= cooc_{t \to t_{co}} \cdot histocc_t$ ▷ $coococc_{t_{co}}$ is initialized with 0 for all t_{co}
  end for
end for

$T_{v_{new}}$, COT_u is calculated for $u = u_{v_{new}}$ and ordered by $coococc_{t_{co}}$, the overall count of the tag t_{co} co-occurring with any tag of the user's history. For all scores $score_{t_{co}}$ $coococc_{t_{co}}$ is normalized by $|\mathcal{V}| \cdot \left|H_{u_{v_{new}}}\right|$ as each tag t can occur at most once per video in $H_{u_{v_{new}}}$ and the number of times t co-occurs with a different tag cannot exceed the number of videos $\mathcal{V}$ considered. This approach, too, fails when a user has not uploaded videos before uploading v_{new}, as then $TH_{u_{v_{new}}}$ cannot be calculated and with this $COT_{u_{v_{new}}}$ cannot be calculated either.

3.5. Channel Based System

Motivation

The History based system, as well as the Co-Occurrence based system, might perform badly if the history of a user is very short, as this might not allow an all too certain prediction of the user's tagging behavior, which might become even more noisy if co-occurring tags are used. Even worse, a user might have no tagging history at all (i.e. he/she has uploaded no videos until now or his/her uploaded videos were not tagged), which results in an empty list of suggested tags, as seen in Section 3.3 and Section 3.4. Especially for these cases an approach is desirable that still produces results and whose results are similarly personalized. This system uses the channels a user has subscribed, which potentially reflect his interest and with this might be a reasonable source for tags that might as well be suitable for the user's own videos. Channels, in this scenario, are the collection of all videos of a user a, the so called author of the channel, that are publicly available. A different user of the same platform may "subscribe" such a channel, meaning that he/she is informed whenever a new video is added to this channel (i.e. the channel's author has uploaded a new publicly available video). For example a music artist might have a channel in which all his music videos are available. The artist's fans can then subscribe to this channel and are informed whenever a new music video is uploaded.

Description

The Channel based system utilizes a user u's subscribed channels CH_u as an exemplary social signal to suggest tags. Each such channel $ch \in CH_u$ is assumed to have a respective

author a_{ch}, who again is a user of the same social platform that u is part of. Although channels are a concept not present on all social sharing websites, this system is in fact easily adjustable to work with every other type of connection between the uploader of a video and other users of the platform (e.g. a is a friend of u, a has posted a comment on content of u, etc.). For this to work, first all authors A_u of the channels of a user u must be collected, which is considered to be easily done by means provided by the social platform. This part potentially has to be adjusted to reflect other types of connection (e.g. collecting all friends of u). Then the tagging history for all authors in A_u is calculated and used to provide the suggested tags, as seen in Algorithm 5. The list of tags $CHT_{u_{v_{new}}}$

Algorithm 5 Construct channel based set of tags for a specific user u

```
CHT_u = ∅
for all a ∈ A_u do
    CHT_u = CHT_u ∪ TH_a
    for all t ∈ TH_a do
        chanocc_t += 1                    ▷ chanocc_t is initialized with 0 for all t
    end for
end for
```

ranked by $chanocc_t$, the number of times the tag t is present in all videos of all channels' authors, is then used for $T_{v_{new}}$, with the respective scores calculated as $\frac{chanocc_t}{\sum_{a \in A_u} |H_a|}$ for each $t \in CHT_{u_{v_{new}}}$, as each tag can occur in at most all videos of every author of the user's subscribed channels. This approach produces an empty $CHT_{u_{v_{new}}}$ if either the user has not subscribed any channels or if every channel subscribed by the user is empty (i.e. the author of each channel has not uploaded any videos).

3.6. Content Based Systems

3.6.1. General Motivation

The general motivation to use content based systems is the hope that the content might hold clues that are not present in any other source of information. A user's history might be highly diverse, the tags co-occurring with these tags might be very noisy and the topics of his subscribed channels might be orthogonal to those of his own uploads. To still get usable information in such a case, the visual signals gained from the uploaded content itself might be a valuable source of information, as the tags gained via these signals are invariant to the user's characteristics, but this also means that they do not personalize at all. Another advantage of such systems might be that they produce tags even if nothing is known about the user, for example if the user has just registered with the platform and is now uploading his first video.

3.6.2. Visual Components

One of the systems that are described in this section uses a so called classification pipeline. Classification in this context means that a set of arbitrary concepts C (containing concepts like "baby" or "airplane-flying") is defined and a classifier is trained which is able to categorize a new video into one of these concepts. The process of such a classification is often called concept detection, as it tries to detect the concept present in the video. For the training of a classifier, a set of videos and their correct concepts need to be known. This set has to be independent from the set of videos that are to be classified and is called the training set V_{train}. This classification pipeline consists of a feature extraction step that describes the videos in a reasonable way and an inference step in which similar videos are used to infer information about v_{new}, like its concept.

Feature Extraction

The feature detection and extraction used in this setting are the SIFT features as introduced in [15] using the implementation of the vlfeat library[1]. These features are invariant to scaling, translation and rotation and robust against lighting changes and affine as well as 3D projection, but it should be remarked that they do not take color into account. As these features work for images only, every video v is represented by a set of keyframes KEY_v. Figure 3.2 shows how the feature extraction used here works.

For all keyframes KEY_{train} of all videos in V_{train}, the SIFT features are extracted, resulting in several so called "patches" for each image. The patches are arranged in the SIFT feature space and clustered using k-means clustering[2]. These clusters are called Visual Words and each is represented by its center. The collection of all k Visual Words is called the codebook. Each keyframe can then be represented by the Visual Words that its patches belong to.

Inference Mechanism

The inference mechanism utilized in this thesis is illustrated in Figure 3.3. To infer the concept of a new video v_{new}, a so called "Nearest Neighbor Classification", similar to the one described in Chapter 4.7 of [8], is used. First the patches of v_{new}'s keyframes must be extracted and each patch is matched to the nearest Visual Word. For each keyframe key the χ^2-distance[3] between its Visual Words representation and the Visual Words representations of all keyframes in KEY_{train} is calculated. Then the k keyframes with the smallest distance, called the k Nearest Neighbors NN_{key}, are considered for inferring the concept of v_{new}. Each keyframe in NN_{key} casts a number of votes for its corresponding video's concept that is the reverse of its rank (e.g. for $|NN_{key}| = 50$ the most similar Nearest Neighbor would cast 50 votes for its video's concept, the second most similar 49

[1] `http://www.vlfeat.org/overview/sift.html`
[2] For more information on k-means clustering see [16].
[3] See [6] for details about this distance measure.

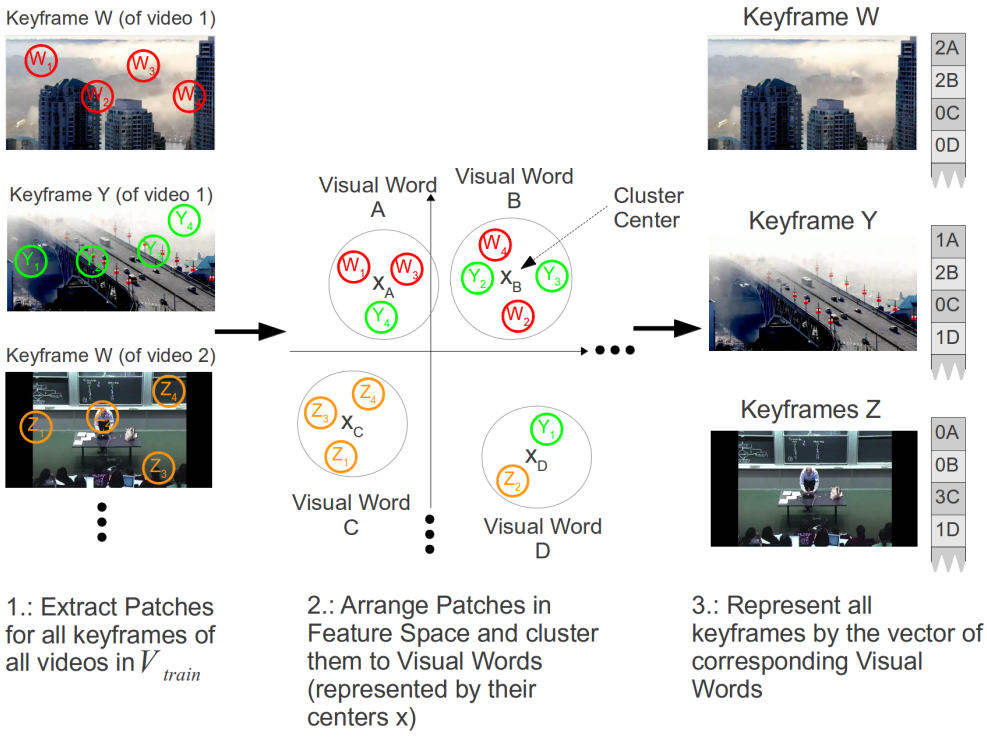


Figure 3.2.: The feature extraction of the concept detection pipeline as used by the Content based systems.

and the least similar only 1). The votes, normalized by the number of all votes $\sum_{i=1}^{|NN_{key}|} i$, are then used as scores for the concepts. To get a decision for v_{new}, each of its keyframes casts a vote for its highest scored concept and v_{new} is assumed to belong to the concept with the highest number of votes.

3.6.3. Concept Vocabulary Approach

Motivation

Using tags from videos that contain the same concept as the newly uploaded one is motivated by the idea that different users might use the same words to described the same concept and with this might use the same tags for videos containing this concept. If for example a user who normally uploads videos concerning firefighters and subscribes channels about kittens uploads a video about skiing, he would not get the correct tags with the systems presented up till now. But if this video's concept, based on the video's visual information, can be detected as `skiing`, tags that are often associated with skiing by other users might be suggested, like `snow`, `skiing` and so on.

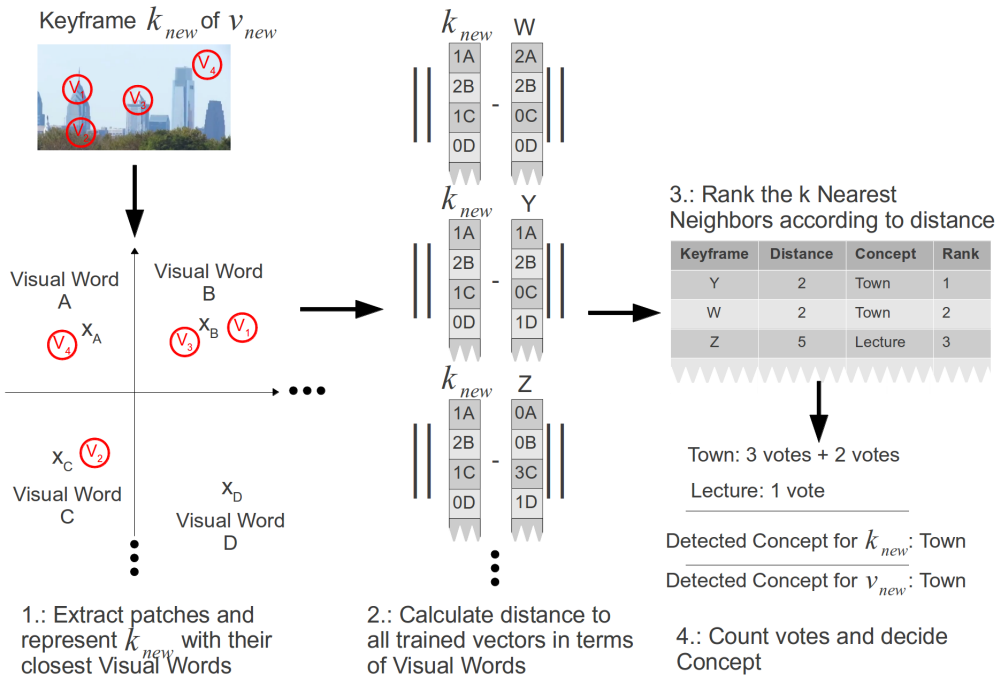


Figure 3.3.: The classification of a new video based on the Visual Words describing it. For this example the video v_{new} is represented by only a single keyframe.

Description

The Concept Vocabulary based approach to tag suggestion uses the concept detection pipeline described in Section 3.6.2 to determine the concept $c_{v_{new}}$ that v_{new} belongs to. Knowing this concept, tags are calculated from other videos $V_{c_{v_{new}}} \subset V_{train}$ in which the same concept $c_{v_{new}}$ is present. This is done by using a concept vocabulary T_c, a subset of the overall tag vocabulary, that consists only of those tags that are used by the videos sharing the same concept V_c. A concept vocabulary for a given concept and a variable that reflects the occurrence of each tag t in this set, denoted as $conocc_t$, are created as shown in Algorithm 6. T_c ordered by $conocc_t$ is then the actual concept vocabulary. To

Algorithm 6 Construct concept vocabulary for a specific concept c

```
T_c = ∅
for all v ∈ V_c do
    T_c = T_c ∪ T_v
    for all t ∈ T_v do
        conocc_t += 1          ▷ conocc_t is initialized with 0 for all t
    end for
end for
```

suggest tags for v_{new} the concept $c_{v_{new}}$ is determined as described above and the ordered tags $T_{c_{v_{new}}}$ are taken as suggested tags for v_{new}. This approach always assigns tags to a video, even if its score for the best concept is low.

3.6.4. Nearest Neighbor Transfer Approach

Motivation

This system is motivated by the idea that the decision for one exclusive concept, as done by the system described in Section 3.6.3, might be faulty, which poses a single point of failure. Furthermore, concept vocabularies might be too noisy or too general to fit the user's needs. Therefore this system tries to utilize the Nearest Neighbors' tags rather than their concept annotations, potentially suggesting tags that belong to several different concepts that might all be present in the video or to a concept that was not trained.

Description

The approach described in this section uses a similar pipeline as the one described in Section 3.6.3. But here the Nearest Neighbors are not used for a classification, but instead the tags are transfered from the Nearest Neighbors directly. To do this, for each keyframe $key \in KEY_v$ of a video v the ordered list of the k best (most similar) Nearest Neighbors NN_{key} is considered. The tags of a Nearest Neighbor $nn \in NN_{key}$ are the ones of its corresponding video and are denoted as T_{nn}. Each Nearest Neighbor gives a number of votes for its tags which is the reverse of its rank in this list. This procedure can be seen in more detail in Algorithm 7. The list of tags TNN_v is then ordered by

Algorithm 7 Transfer tags from Nearest Neighbors for a specific video v

```
TNN_v = ∅
for all key ∈ KEY_v do
    for all nn ∈ NN_key do
        TNN_v = TNN_v ∪ T_nn
        for all t ∈ T_nn do
            invrank = |NN_key| − rank(nn)        ▷ rank starts with rank 0
            nnocc_t += invrank                   ▷ nnocc_t is initialized with 0 for all t
        end for
    end for
end for
```

the number of votes a specific tag t received in total, denoted as $nnocc_t$. For $v = v_{new}$ the ordered $TNN_{v_{new}}$ is the list of tags suggested for v_{new}. To attain the scores, $nnocc$ is normalized by $|KEY_{v_{new}}| \cdot \sum_{i=1}^{|NN_{key}|} i$ for every t, as a tag can get at most the votes (whose number decreases with rank) of all Nearest Neighbors and this can happen for at most all keyframes associated with the video.

3.7. Visual Personalized Tag Transfer

Motivation

The Visual Personalized Tag Transfer based system is motivated by the fact that merging multiple results might have the benefit of reducing noise and producing more reliable results. A general example for this can be seen in Figure 3.6 in Section 3.8. The system presented in the following is able to merge results gained from the Nearest Neighbor Transfer and History based approach. In a special case it becomes a modified version of the History based system which does not treat every video in the user's history the same, but rather weights them depending on the visual similarity to the newly uploaded video. This might be especially beneficial for expressive but non-coherent user histories, as can be seen in the example in Figure 3.4.

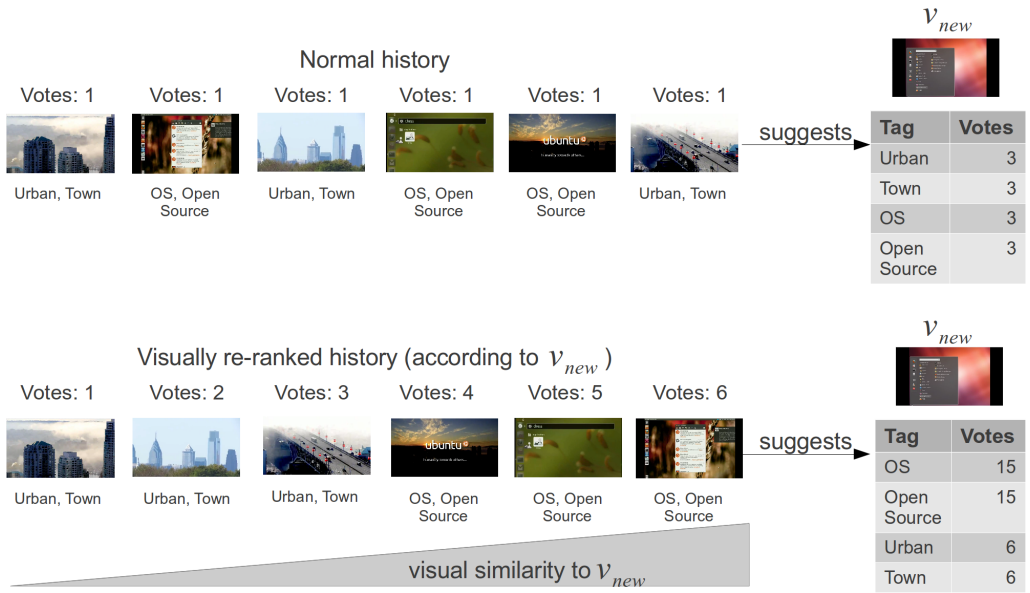


Figure 3.4.: An example for the benefits of visually re-ranking the history according to v_{new}.

Description

Much like for the Nearest Neighbor Transfer approach, the video v is split in keyframes KEY_v representing it. The Visual Personalized Tag Transfer system consists of two parts:

- The set of k Visual Nearest Neighbors NN_{key} (with the corresponding distances) of a keyframe $key \in KEY_v$, as gained from the Nearest Neighbor Transfer approach.
- The videos in the history of the uploading user H_{u_v} which are each represented by a single keyframe.

The mechanism of the Visual Personalized Tag Transfer system for one keyframe $key \in KEY_v$ is visualized in Figure 3.5. As can be seen there, the history is visually re-ranked

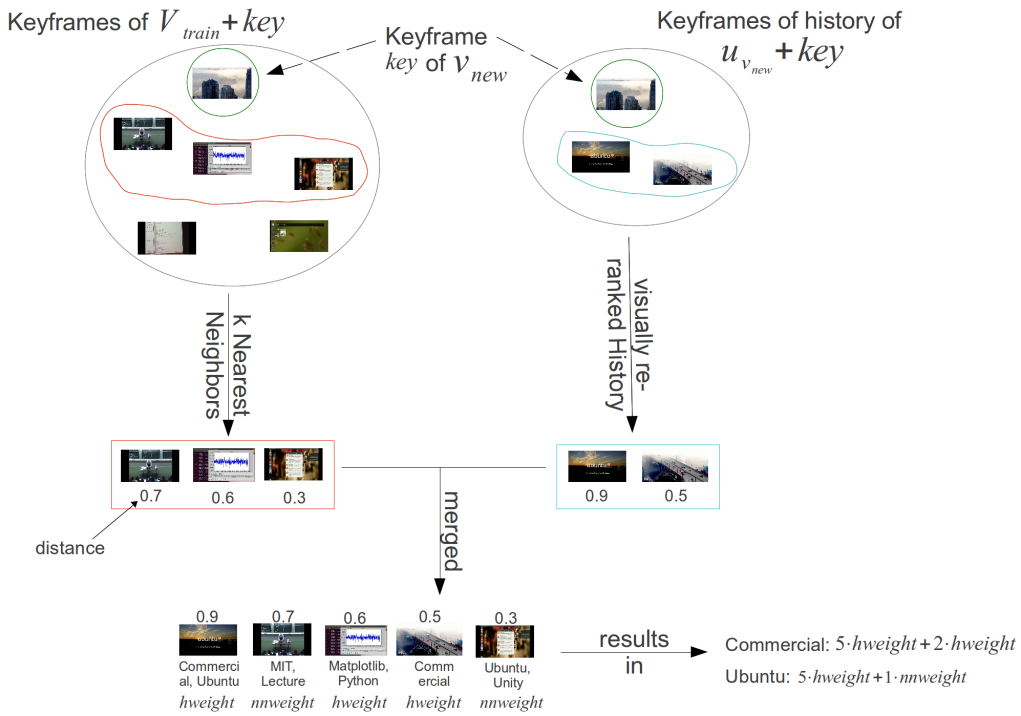


Figure 3.5.: The working mechanisms of the Visual Personalized Tag Transfer based tag suggestion system for a keyframe $key \in KEY_{v_{new}}$.

in respect to *key*. This means that for every video in H_{u_v} the visual distance (as defined in Section 3.6.2) between its keyframe and *key* is calculated and assigned to this video. The videos ranked by these distances (the smallest distance has the highest rank) is then the visually re-ranked History. The merged list of videos MNN_{key} is the unification of NN_{key} and the re-ranked H_{u_v} and is ordered by the respective distances to *key* (again the smallest at the top).

To allow to adjust the degree of personalization (i.e. how strongly the user's history influences the result), the personalization fraction *perfrac* is introduced. It is a number between 0 and 1 and influences the ratio between the two weights *hweight* (the weight for a video $v \in H_{u_v}$) and *nnweight* ($v \in NN_{key}$). As the number of videos in H_{u_v} can be different from the number of videos in NN_{key}, the weights have to compensate this. If for example H_{u_v} consists of 4 videos and NN_{key} of 8 and both should have the same influence (i.e. $perfrac = 0.5$), then $hweight = 2$ and $nnweight = 1$ must hold for the minimal integer case. This is achieved if the weights for a given *perfrac* are calculated such that $perfrac = \frac{|H_{u_v}| \cdot hweight}{|NN_{key}| \cdot nnweight}$ holds and both weights are minimal integers (in a real implementation the accuracy has to be limited to get reasonably sized weights). For $perfrac = 0$ *hweight* is 0 and *nnweight* is 1 and therefore the Visual Personalized Tag Transfer system's performance is identical to the performance of the Nearest Neighbor Transfer approach. For $perfrac = 1$ *hweight* is 1 and *nnweight* is 0 and the corner case described in the Description and in Figure 3.4 is reached. The number of votes that each video v in MNN_{key} casts for its tags is the product of its inverse rank in MNN_{key}

and either *hweight* or *nnweight*. The number of votes a tag gets is summed up over all $key \in KEY_v$. This can be seen in more detail in Algorithm 8.

Algorithm 8 Transfer tags from mixed Nearest Neighbors for a specific video v

```
TMNN_v = ∅
for all key ∈ KEY_v do
    for all nn ∈ MNN_key do
        TMNN_v = TMNN_v ∪ T_nn
        for all t ∈ T_nn do
            invrank = |MNN_key| − rank(nn)        ▷ rank starts with rank 0
            if nn ∈ H_uv then
                fweight = invrank · hweight
            else
                fweight = invrank · nnweight
            end if
            mnnocc_t += fweight                   ▷ mnnocc_t is initialized with 0 for all t
        end for
    end for
end for
```

$TMNN_v$ (the set of all tags found by this approach) ordered by $mnnocc_t$ (the votes that a tag t has received) results in the final list of tags that is suggested by this approach. The scores can be calculated by normalizing $mnnocc_t$ by the total number of votes cast.

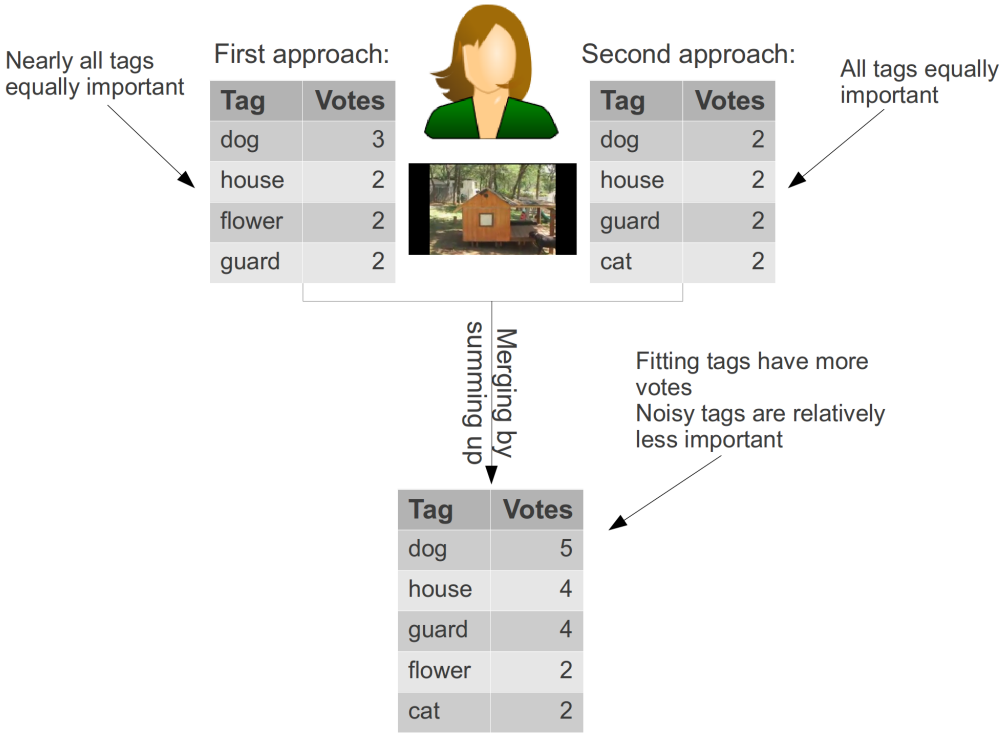
3.8. Fusion

This section deals with several ways of combining or "fusing" some of the tag suggestion systems described in the previous sections. The fusion of systems, as described in the following, is again a tag suggestion system of its own, defined in Section 3.1. Fusion in this context means that results of the subsystems (these can be intermediary results as well as the final list of tags and their scores) are combined in a manner that the combination of these information streams is again usable for suggesting tags. Two fusion systems are described in the following sections. These are (in order) a rule based system, which uses a rather simple fixed rule to fuse three of the systems described up to this point. And a more sophisticated system that uses a weighted sum between the lists of tags and their respective scores or ranks computed by four of the described tag suggestion systems, using either global weights or individual weights.

General Motivation

The motivation behind fusing several tag suggestion systems into one is to eliminate the weaknesses of the single subsystems. When combining multiple systems, it is possible that one of the subsystems performs good in cases where the other subsystems do not

and would fail on their own. Especially, as most of the systems have special cases in which they are not able to suggest tags at all, it is desirable to combine multiple systems so that always at least some tags are suggested. Furthermore, even if all systems would perform acceptable on their own for a given user and his newly uploaded video, the combination of them might be able to reduce noise and suggest more accurate tags. A general example for this can be seen in Figure 3.6, this is only to visualize the theoretical benefit and does not depict the mechanism of an actual system. It is unlikely that all considered systems produce the same kind of noise (as the set of unused tags is fairly large) but for acceptable systems it is likely that they suggest the same correct tags (as the set of correct tags is rather small in comparison).



Tag	Votes
dog	3
house	2
flower	2
guard	2

Tag	Votes
dog	2
house	2
guard	2
cat	2

Tag	Votes
dog	5
house	4
guard	4
flower	2
cat	2

Figure 3.6.: Illustration of the benefits of merging, in terms of noise reduction. This does not depict an actual system.

3.8.1. Rule Based Fusion

Motivation

The Rule based fusion tries to mimic an expert manually combining the systems based on knowledge about the specific video, its user and his/her channels. This is done by using fixed rules based on the performances of the systems it fuses. It is introduced as a baseline for fusion systems and was mainly used to gain first insights about the capabilities of fusion based systems. Furthermore, it has only two easily tunable (e.g. by a grid search on a very limited range) parameters.

A grid search is often used when a number of parameters have to be tuned in respect to a certain performance measure (here averaged Precision/Recall). Every possible combination of the parameters in a certain interval, discretized by a variable called the "step", is tested with the real systems and evaluated in terms of the aforementioned performance measure. The results of this procedure are those combinations of parameters that have performed best.

The Rule based fusion builds on a presumed correlation between the History or Channel size and the expressiveness of the History or Channel based tag suggestion system respectively, to dynamically decide which system to use.

Description

The Rule based fusion system utilizes knowledge about the uploading user, which is readily available from the platform, to choose one of three systems. These three systems are:

- The History based system as described in Section 3.3, with its resulting ordered list of tags TH_u.
- The Channel based system, with the corresponding ordered CHT_u, as seen in Section 3.5.
- The system that utilizes visual Nearest Neighbors for tag suggestion, which was described in Section 3.6.4 and is associated with its ordered list TNN_v.

The Rule based system has two statically chosen parameters that influence its performance, namely *minhist* and *minchan*, both positive integers, and works as described in Algorithm 9. This algorithm shows that this system simply tests if the user's History is large enough ($\geq$ *minhist*), if so it uses the History based system's results. If not, the number of Channels is compared to *minchan*. If it is larger or equal, then the list of tags as provided by the Channel based system is used. If not, the tags that are suggested by the Nearest Neighbor Transfer based approach are suggested. The list of tags that is

Algorithm 9 Rule based fusion for a specific video v

if $|H_{u_v}| \geq minhist$ **then**
  $RT = TH_{u_v}$
else if $|CH_{u_v}| \geq minchan$ **then**
  $RT = CHT_{u_v}$
else
  $RT = TNN_v$
end if

suggested by this system is then RT (for a video $v = v_{new}$) as described before and is equal to one of the three lists that would have been provided by the History, the Channel or the Nearest Neighbor based system. Which of these three is chosen for a specific video depends on the two parameters *minhist* and *minchan*, as well as on the uploading user's

specifics and because of the latter this system is actually a personalizing one. For extreme cases of *minhist* and *minchan* this system is independent of the user: For $minhist = 0$ it behaves just like the History based system, for $minhist = \infty$ and $minchan = 0$ it behaves like the Channel based system and for $minhist = \infty$ and $minchan = \infty$ it behaves like the Nearest Neighbor Transfer approach. This system is likely to provide tags for $minhist > 0$ and $minchan > 0$, as at least $TNN_{v_{new}}$ is normally non-empty and if one of the other lists of tags is chosen, these are less likely to be empty, as at least one previously uploaded video exists or a subscribed channel respectively, although these might have no tags.

3.8.2. Weighted Sum Based Fusion

Motivation

One motivation for this approach to fusion is to gain further insights regarding the capabilities of fusion in general. This is especially true for this system, as each of its parameters directly corresponds to the influence of one of the subsystems. Another motivation is to propose a fusion framework that is easily extensible without becoming overly complex, but is still more capable than the Rule based system. Furthermore, a system might be advantageous that does not depend on static rules but can learn a specific behavior for each user. This can be implemented with weights that are learned for each user.

Description

The Weighted Sum based fusion uses the weighted sum[4] of the results of its subsystems – a subset of the tag suggestion systems described up till now, denoted as SYS – to generate its own list of suggested tags. The subsystems used here are:

- The History based system *hist* (as seen in Section 3.3)
- The Visual Personalized Tag Transfer system with $perfrac = \alpha$ (described in Section 3.7), denoted as $ptt(\alpha)$.
- The Channel based system *chan* (see Section 3.5).
- The system that uses Co-Occurrence as its modality *cooc* (Section 3.4).

LT_{SYS} is the collection of all lists of tags suggested by those systems T_{hist}, $T_{ptt(\alpha)}$, T_{chan} and T_{cooc}. Every subsystem $sys \in SYS$ is assigned a weight w_{sys} that indicates its influence on the final result. If this weight is zero, the system's tags will not be considered at all, allowing to completely ignore them (which might help to reduce noise). Furthermore, every subsystem has a *rating* for each tag in its list of suggested tags T_{sys} which can either be the tag's inverse rank in this list or the tag's normalized score $score_{sys}(t)$, as defined in each system's description. Each system gives a number of votes that is equal to $w_{sys} \cdot rating$. The votes a tag t gets are summed up over all system, resulting in the

[4]Information on general weighted calculus can be found in Chapter 2 of [17], more specific information regarding the Weighted Sum model in decision making can be found in Section 6.2 of [5].

so called weighted sum ws_t. This is illustrated in Algorithm 10 (for readability's sake, the list of tags of a system *sys* that operates on a user rather than on a video is still denoted as T_{sys_v}, implying $T_{sys_{u_v}}$). TWS_v – the set of all tags that occurred in at least

Algorithm 10 Weighted Sum fusion for a specific video v

$TWS_v = \varnothing$
for all $T_{sys} \in LT_{SYS}$ **do**
  if $w_{sys} > 0$ **then**
    $TWS_v = TWS_v \cup T_{sys_v}$
    for all $t \in T_{sys_v}$ **do**
      if $rank ==$ `True` **then** ▷ Use rank for rating
        $rating = \frac{1}{rank(t)}$
      else ▷ Use (normalized) score as rating
        $rating = score_{sys}(t)$
      end if
      $ws_t \mathrel{+}= w_{sys} \cdot rating$ ▷ ws_t is initialized with 0 for all t
    end for
  end if
end for

one $T_{sys} \in LT_{SYS}$ – ordered by ws_t is then the list of tags that this fusion system suggests. If one or more systems get a weight of zero, the Weighted Sum fusion system behaves like the fusion of just those systems that have a non-zero weight. If only one system has a non-zero weight, the fusion will suggest the same tags as this system. Should all systems get a zero weight, then this fusion will return an empty list of suggested tags. If at least one of the systems with a non-zero weight produces tags for a given video, the fusion will provide tags as well. The choice of α has the influences as described in Section 3.7.

4. Experiments

In this Chapter the systems and the fusions of these systems, as introduced in Chapter 3, will be evaluated on a real life setup. First, the test setup will be characterized, including the choice of the social platform, the metrics of evaluation, the crawling of data and the composition of the training and test sets, as well as the limitations due to the underlying social platform. The next sections will then evaluate and discuss the systems in the same order as in Chapter 3. This chapter ends with a comparison of the performances of the systems and their fusions.

4.1. Test Setup

For all following tests, YouTube[1] will be considered the underlying social platform. The resources will be videos uploaded to YouTube. A choice that is supported by the fact that YouTube is the major video sharing platform in the Internet (compare Chapter 1) and with this provides a huge amount of realistic data, including not only videos but also metadata, like the uploading user, the tags associated with the video and several types of connection between users and videos.

4.1.1. Dataset

The dataset is based to a wide extend on the dataset kindly provided by Markus Koch. Because of this, his master's thesis that originally worked with this data should be consulted for details on the actual crawling of the data[2]. The dataset is divided in 230 concepts (e.g. `boxing` and `drawing`) which consist of about 200 videos each. For a list of all concepts and the queries used to find the videos see Table A.1. About 50 videos per concept have been downloaded in addition to those provided by Markus Koch, using youtube-dl[3], to increase the size of the corpus in order to get more reliable results from the Nearest Neighbor based approaches. For the keyframe extraction (needed for the systems described in Section 3.6.2) of these additional videos an algorithm was used that detects changes between frames. It calculates the difference in pixel values and only extracts new frames if the change between the last extracted frame and the one considered for extraction surpasses a certain threshold. The actual number of keyframes extracted by this varies with the length of the video and the amount and degree of change happening

[1] `www.youtube.com`
[2] See Chapter 6.1 of [11].
[3] http://rg3.github.com/youtube-dl/

in it. Together with the keyframes extracted for [11] this resulted in roughly 3 million keyframes. Furthermore, for each video that was present in a user's history, a single keyframe was downloaded from YouTube[4] which is the image that is used on YouTube to provide the thumbnail for the video.

For each video additional meta information has been crawled, namely the user who uploaded the video and the actual tags he used for this video on YouTube. In addition to this, the user's history (i.e. the list of videos uploaded, excluding the original video), including the tags for each video in the history, were stored. Furthermore, the user's subscribed channels, together with the videos and tags in the history of each channel's author, were crawled. All this was done using the python-bindings for the official YouTube Data API[5].

4.1.2. YouTube's Structure and Limitations

On YouTube, every video is uploaded by a unique and unambiguously identifiable user. For each user a list of the videos uploaded by him/her up till now is available, called the history. Furthermore, the list of channels subscribed by a user can be accessed over the Data API. Each entry in this list of channels is associated with a YouTube user, who is the channel's author. All videos are associated with a list of tags. These relationships are illustrated in Figure 4.1.

There are, however, some limitations regarding the access to this information. A user's history as available through the Data API is capped at the 25 most recent videos, but this kind of restriction would have been made regardless, to save computational time. Furthermore, a lot of social information is either not available through the API or is configurable to be private. This mainly affects the social signals, as they cannot be used for users who have configured the respective source of information to be private. Because of this, Channels were chosen as a source of social signals, as these showed a good compromise between expressiveness and availability, although only ~44% of the users' channels were publicly available. Another limitation that is inherent to YouTube is the deletion of videos (e.g. because of copyright infringements) and user accounts (issued by the user or YouTube). As this thesis builds on the dataset provided by [11], and as [11] has been written in 2011, this prevented the crawling of the additional information needed for this thesis (i.e. channel and history information) of a number of videos which had already been deleted. Therefore, two concepts are represented by less than 200 videos, namely `santa` (110 videos) and `golf course` (150 videos).

4.1.3. Setup and Metrics

The concept detection is trained on a set of videos called the training set V_{train}, a subset of the dataset. V_{train} consists of 100 randomly chosen videos per concept, resulting

[4] Available via the url `http://img.youtube.com/vi/VIDEOID/0.jpg`

[5] `https://developers.google.com/youtube/1.0/developers_guide_python`

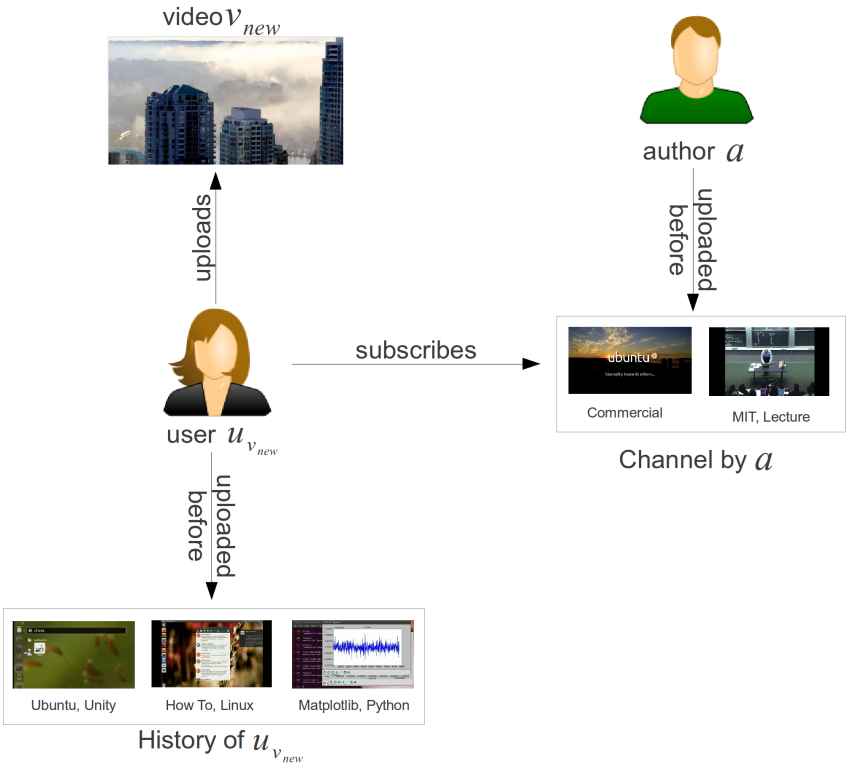


Figure 4.1.: The structure of YouTube as used by the tag suggestion systems described in this thesis.

in about 23,000 videos (~50% of the dataset). The tag suggestion systems and the concept detection itself are evaluated on a set of videos that is called the testing set V_{test}, with V_{train} and V_{test} being disjunct. V_{test} consists of the remaining videos for each concept (about 100, except for the two concepts discussed in Section 4.1.2). Furthermore, the number of keyframes is reduced to 1,000 per concept, resulting in roughly 230,000 keyframes for each train and test set. For this, all extracted keyframes are randomly subsampled, meaning that videos, that were represented by more keyframes than the others before reducing are more likely to be represented with many keyframes afterwards. The randomness of this sampling is only restricted by the fact that each video has to be represented by at least one keyframe.

All tags of all videos, also called the tag vocabulary, are only reduced by a number of Stop Words (initial list kindly provided by Damian Borth, for a full list see Table A.2). Stop Words are words that are considered to carry only little information, especially articles (like `a` or `the`) and particles (e.g. `of` or `on`), and are thus removed in every video's list of tags and with this are never suggested by any tag suggestion system.

As stated in the Introduction, the performance of the systems is evaluated on the tags that were assigned by the original uploader on YouTube. One should keep in mind that these metrics only evaluate whether a tag suggestion system is able to suggest tags that a user actually used. This might be a good way to catch tags with a personal meaning for the user (see Section 1) but this also means that additional possibly fitting tags that the

user did not use are not considered in the numbers presented in the following. Although manual evaluation, e.g. by user studies, could evaluate unused but fitting tags, it is easily deluded by too general terms and might not catch the uploader's intentions (e.g. someone who evaluates a video but has never played a first person shooter might not recognize tags like `fps` or `frag` as correct). Therefore it is left out in this thesis.

The tags provided by a video v's original uploader u_v are called the ground truth GT_v. The metrics that will be applied for evaluating the performance of the systems are the Precision P and the Recall R. The Precision indicates what part of the suggested tags T_v is correct (in respect to the ground truth) and is calculated as $P_v = \frac{|T_v \cap GT_v|}{|T_v|}$. The Recall on the other hand is a measurement for how many of the tags that were used by the uploader were actually suggested by the system, this is calculated as $R_v = \frac{|T_v \cap GT_v|}{|GT_v|}$. The concept of Precision and Recall is also illustrated in Figure 4.2. For the purposes of this thesis, we restrict the list of tags that the systems suggest to a maximum number of tags N, as motivated in Section 3.1, and therefore calculate the so called Precision@rank or, in this case, Precision@N, denoted as $P@N$, meaning that only the first N elements of T_v are considered. This is analogous for Recall@rank ($R@N$). If a video has no tags or only tags that are Stop Words, $R@N = P@N = 0$ is chosen, the most pessimistic measure.

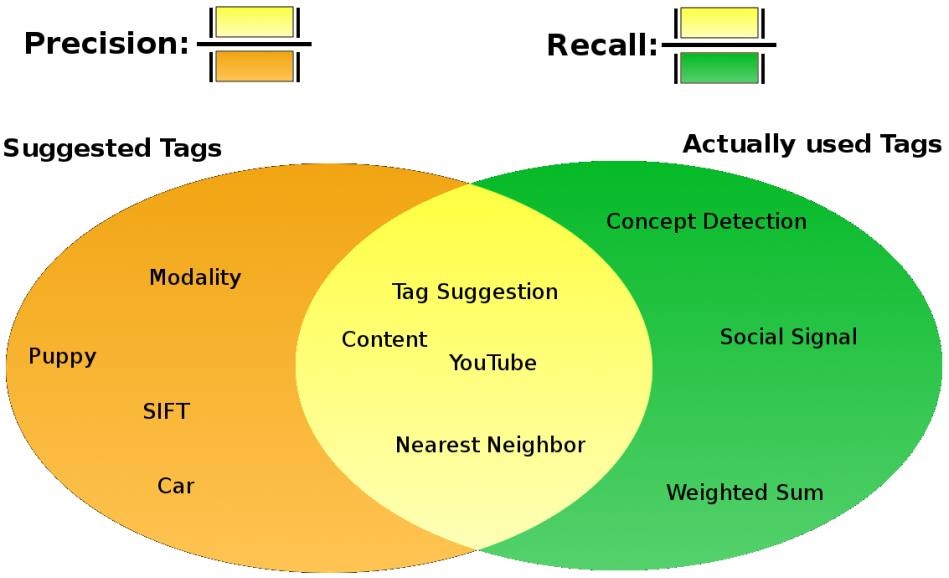


Figure 4.2.: Precision and Recall, two measurements for the performance of a tag suggestion system.

As this thesis considers a rather large set of videos, in most cases the averaged Precision@rank ($AP@N$) and the averaged Recall@rank ($AR@N$) will be considered, meaning that the Precision, and Recall respectively, is calculated for every video $v \in V_{test}$ and the average of these values is considered. As there is always a trade-off between AP and AR (see [30], which also contains a more comprehensive review on these metrics, as well as the Average Precision which should not be confused with the averaged Precision), in addition

to these measures a figure will be presented for most systems that plots AR against AP. The points in this plot represent a value of N from 1 (leftmost) to 25 (rightmost), the x value of a point reflects its $AR@N$ for this N and the y value is the $AP@N$. These points are connected with a line to better illustrate potential trends. In addition to the individual plots, there is a comparative section in this chapter that separately compares the averaged Precision and the averaged Recall.

4.2. Global Tag Statistic Based System

To better understand the results provided by this system, first the tag vocabulary, built on the tags used by all video in V_{test}, will be examined. In total 246,591 different tags that were not Stop Words are used. The top 25 of which can be seen in Table 4.1. These are the tags that this system suggests for $N = 25$ for every video, independent of its content or the user that uploaded it. The figure plotting $AR@N$ against $AP@N$ is Figure 4.3 and contains the plot as described in 4.1.3. As can be seen there, this system performs poorly in terms of both $AP@N$ and $AR@N$, with a maximum of a bit less than 0.032 for $AP@N$ and a bit more than 0.034 for $AR@N$. A fact that confirms the assumption that this system is a reasonable lower bound.

Table 4.1.: The top 25 most used tags in V_{test}, together with the number of times they were used and the number of users who used this tag.

Tag	**Usage**	**Users**	**Tag**	**Usage**	**Users**
travel	3736	2174	city	1832	1154
music	3163	2133	news	1810	872
world	2978	1896	tour	1795	1141
funny	2876	1943	hd	1792	1074
vacation	2775	657	tripadvisor	1749	12
show	2475	1256	tripwow	1749	11
trip	2422	477	racing	1704	940
car	2358	1425	water	1702	1227
nature	2272	1431	beach	1696	1192
photography	2191	352	adventure	1671	1040
park	2057	1269	food	1670	765
slideshow	1935	163	big	1662	1131
photos	1842	150			

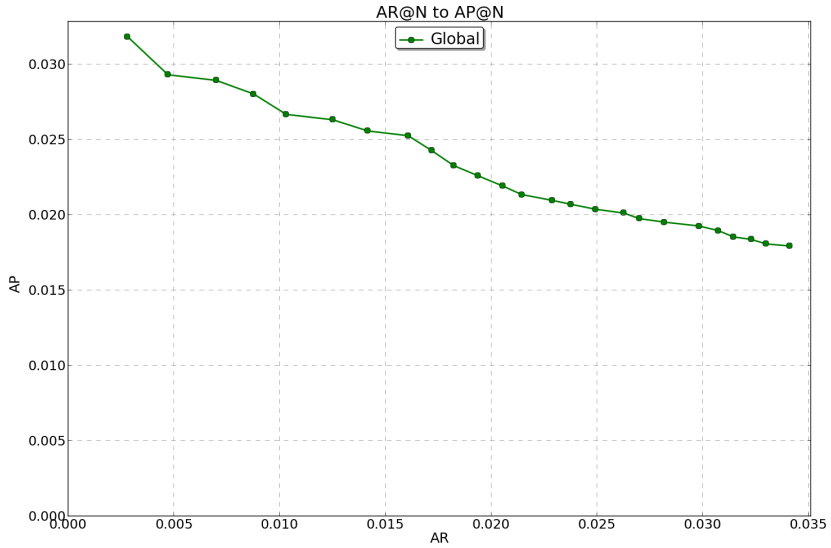


Figure 4.3.: The averaged Precision@N plotted against the averaged Recall@N, for $N \in \{1, \ldots, 25\}$ (left to right), both of the Global Tag Statistic based tag suggestion system.

4.3. History Based System

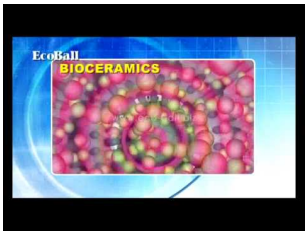
As this system does not suggest any tags if no history is available, it is of interest how the number of videos in the users' histories (also called the length of the history) is distributed. This is illustrated in Figure 4.5 (note the logarithmic scale). The beginning of the curve seems to imply a power law distribution, but peaks at 24 and 25. This can be explained by the fact that the YouTube API only shows the most recent 25 videos when looking at a user's videos (see Section 4.1.2), effectively adding up the whole "tail" of the assumed power law distribution at 25 (and 24 for those users, for whom the video in V_{test} is part of the most recent 25 videos and had to be excluded from the history). It can be seen that there are in fact more than 1,000 users without a history, but these are only about 5% of all users in V_{test}. This suggests that the History based system is not easily surpassed by just suggesting tags for those users that have no history, but that a better system must also suggest better tags for the other users as well. The performance of the system is illustrated in Figure 4.6. It shows that this system has an averaged Precision of nearly 50% for $N = 1$ and for $N = 10$ in average a fourth of the suggested tags is correct. This system has an averaged Recall of about 22.5% for 10 tags. The maximum averaged Recall in this interval is more than 30% for 25 suggested tags. These numbers support using this system as a baseline for the proposed systems, as they show that this system is already quite capable and could be used in real life situations.

Figure 4.4 shows an example result. Each image represents a video uploaded on YouTube. The leftmost image is an example for Precision@6 between 1 and 0.8 and the rightmost for Precision@6 between 0.1 and 0. The tags of the image in the middle achieved Precision@6 in the range of averaged Precision@6$\pm$0.05. Depicted under each image are the tags used

by the video's uploader (preceded by "U:") and the tags suggested by the History based system (preceded by "S:"). Tags in green are guessed correctly, tags in red are Stop Words. The uploader of the top video has the maximum of 25 videos in the history and used a total of 201 tags (40 unique) for those, whereas the bottom video's uploader has only a single video in the history with only 3 tags (and can therefore only be suggested 3 tags instead of 6).

U: devon energy, devon tower, oklahoma city, okc, downtown
S: oklahoma city, okc, oklahoma, downtown, devon tower, devon energy

U: mount, everest, maldives, himalayan, dreams
S: himalaya, nepal, tibet, himalayan, dreams, hollywood



U: washing ball, eco laundry ball, wash ball
S: bola pencuci ajaib, cara mencuci pakaian, tips mencuci pakaian

Figure 4.4.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by the History based system. Precision@6 is from left to right: top values, near AP@6, bottom values.

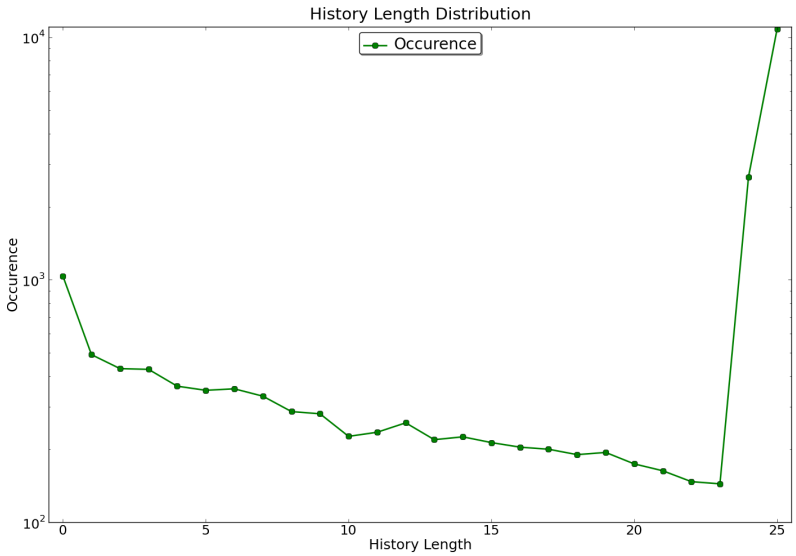


Figure 4.5.: Plotting the history length (the number of videos in the history of a specific user) against the number of occurrences of this length in the test set.

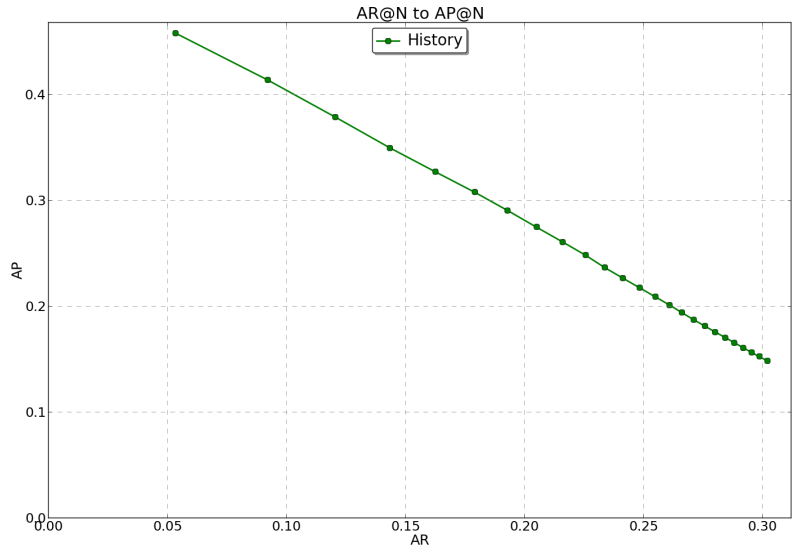


Figure 4.6.: The averaged Precision@N plotted against the averaged Recall@N, for $N \in \{1, \ldots, 25\}$ (left to right), both of the History based tag suggestion system.

4.4. Co-Occurrence Based System

Example results for the Co-Occurrence based system can be seen in Figure 4.7. The worst performing of the three examples has a very diverse history (a total of 43 tags with 39 unique tags, with tags like `skydiving`, `roller coaster`, `dance`, `gay` and `cat`) and with this diverse co-occurring tags. The best performing example is a local Orlando, Florida TV channel that mostly tags its history videos with its name and the weather related videos with additional weather related tags. This system performs considerably worse than the History based one, but much better than the one using a Global Tag Statistic. This can be seen in more detail in Figure 4.8. The highest ranked tag is correct in only about 15.7% of the cases and even when suggesting the maximum of 25 tags, only a Recall of about 13.3% is achieved. This might be, amongst other things, due to the fact that this system is explorative in nature but is evaluated on a measure that does not necessarily reward such behavior, especially not for consistently tagging users.

U: weather, forecast, central, florida, wesh, orlando
S: forecast, central, florida, weather, orlando, wesh

U: high, stakes, poker, daniel, negreanu, vs, dwan
S: high, hd, poker, definition, stakes, school

U: hiking, catskill, hike, waterfall
S: music, comedy, funny, travel, park, water

Figure 4.7.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by the Co-Occurrence based system. Precision@6 is from left to right: top values, near AP@6, bottom values.

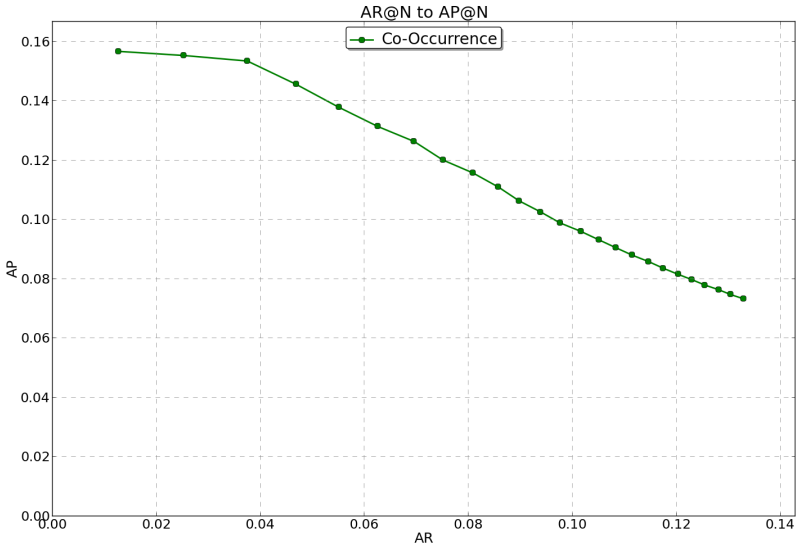


Figure 4.8.: The averaged Precision@N plotted against the averaged Recall@N, for $N \in \{1, \ldots, 25\}$ (left to right), both of the Co-Occurrence based tag suggestion system.

4.5. Channel Based System

Only about 44% of the users make their channel information public, therefore it is necessary to consider two different scenarios for evaluation. The first is to evaluate this system only on those users with public channels. This might be unfair, as the API does not allow to distinguish between a user who has no channels and one who has set them to private and with this, this approach might give an overly positive impression of the availability of channels. This approach should therefore be considered an upper bound for a Channel based system.

The other scenario is to evaluate the system on all users and let the system suggest no tags for those users without (accessible) channels. This might be unfair as well, as not all users whose channels were not accessible have to have no channels, but might just have set this information to private, and with this, this approach should be considered a lower bound.

To reflect this, there is a curve in figures 4.10 and 4.11 for each approach. For better visibility, averaged Precision and averaged Recall are split into two figures, Figure 4.10 ($AP@N$) and Figure 4.11 ($AR@N$). It can be seen that the value for the averaged Precision@1 for the real system is between about 8% and 18%, as these are the lower and upper bound respectively. The maximum averaged Recall (@25) is between 4.1% and 9.2%. These numbers show that the performance of this system must be considered mediocre at best, especially with the performance of the History based system in mind. Example results for the more pessimistic approach can be seen in Figure 4.9. The user of the better performing example has subscribed only a single channel whose author uses the same tags for all videos. The user of the worse performing example has subscribed

5 channels whose authors have very different tags and use most tags very infrequently. One author uses his own username for all his videos.

U: golf, course, yardage, book, design, guide, greens, putting, fly-over, stracka.com
S: golf, greens, yardage, course, fly-over, book

U: eagle, tennis, ball, white, tailed, bird, of, prey, talons
S: xnaruhina, ost, hetalia, xhitsuhina, studios, otaku

Figure 4.9.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by the Channel based system. Precision@6 is from left to right: top values, bottom values. The middle image is omitted as for this systems values around AP@6 are the same as the bottom values.

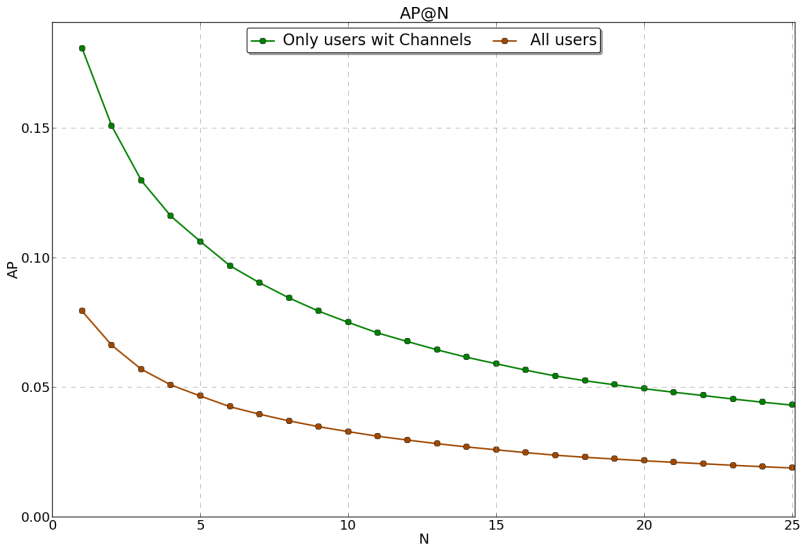


Figure 4.10.: The averaged Precision@N for $N \in \{1, \ldots, 25\}$ of the Channel based tag suggestion system. Comparing the system's upper (only users with Channels) and lower (all users) bounds.

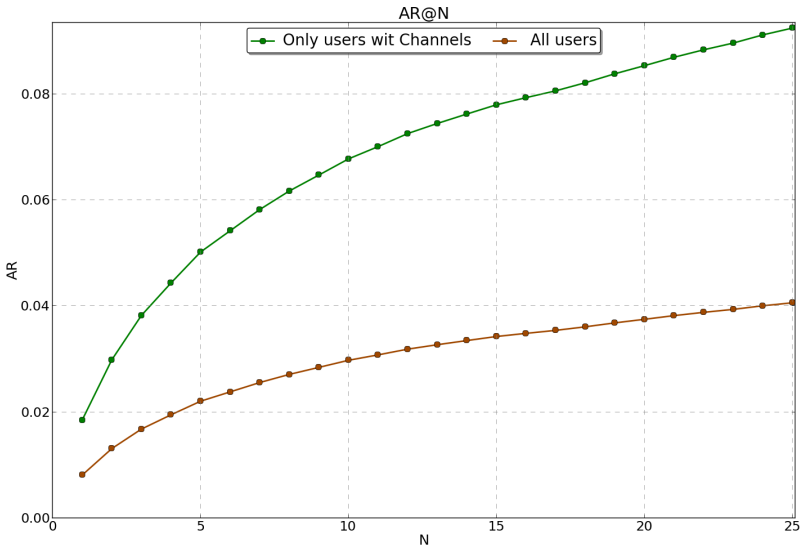


Figure 4.11.: The averaged Recall@N for $N \in \{1, \ldots, 25\}$ of the Channel based tag suggestion system. Comparing the system's upper (only users with Channels) and lower (all users) bounds.

4.6. Content Based Systems

In this section the two systems based on the content signals will be evaluated. As they depend on the underlying concept detection pipeline, as described in Section 3.6.2, its capabilities will also be shown. Furthermore, a so called "oracle" system will be described and evaluated which assumes a concept detection pipeline that always finds the correct concept for a video (i.e. the concept it was crawled for).

4.6.1. Concept Detection Pipeline

The concept detection pipeline can be evaluated using Average Precision. It evaluates how many of the videos that were recognized as a certain concept actually belong to that concept. The Precision P for a single video can either be 1 (correct concept) or 0 (wrong concept). For a concept c, the videos that actually belong to this concept are denoted as V_c and are considered the ground truth. The videos that are detected by the pipeline to belong to c are denoted as $V_{d(c)}$. With this the Average Precision is calculated as $AP = \frac{|V_{d(c)} \cap V_c|}{|V_{d(c)}|}$. If the mean over all Average Precisions of all 230 concepts is taken, this is called the Mean Average Precision MAP. For the concept detection pipeline used in this thesis the following parameters are chosen: 500,000 patches are extracted, 3,000 clusters are created (resulting in a codebook consisting of 3,000 Visual Words) and all images are scaled to a resolution of 250 by 250 pixels. The patches were sampled using a so called multiscale sampling, meaning that the sampling is done several times and the patch size (and therefore also the step) is multiplied (scaled) with a different factor each time. For the Nearest Neighbor search a hash based approximation is used that finds $k = 100$ Nearest Neighbors. This results in a MAP of about 5.33%. The single Average Precisions for the concepts can be seen in Table A.3. This low accuracy indicates that solely depending on the concept detection can be suboptimal.

4.6.2. Concept Vocabulary Approach

This section will evaluate the Concept Vocabulary Approach described in Section 3.6.3. For this the real implementation is evaluated as well as an "oracle" system. For the oracle system, the concept vocabulary is built just like the normal system. But instead of using a real concept detection pipeline to determine to which concept the video belongs, it uses the concept the video was crawled for. This, of course, is not a system applicable to a real life scenario as it uses information that would not be available there, but it shows how a real system could perform with an underlying perfect, or at least very good, concept detection pipeline. This is especially interesting when considering that the underlying concept detection for the real system does not perform all too well, as can be seen in Section 4.6.1.

To allow a better comparison between these two instances of the Concept Vocabulary based tag suggestion system, their performance in terms of averaged Precision@N and

averaged Recall@N is plotted in one figure respectively. These figures are Figure 4.12 and Figure 4.13. Again, for more clarity, these plots are not combined into a single one, but rather one for each measurement is shown. Here, it can be seen that the oracle system clearly outperforms the realistic system. The most relevant tag is correct in more than 60% of the cases for the oracle system, whereas this is only less than 10% for the real system. Furthermore, the oracle system is able to suggest over a fourth of the tags the user used for $N = 25$ – the real system is only able to suggest a bit more than 5%. This further fortifies the assumption that relying on the concept detection alone might not be the best way of utilizing the content signals gained from the videos. The difference in performance is easily explained by the rather poor performance of the concept detection pipeline.

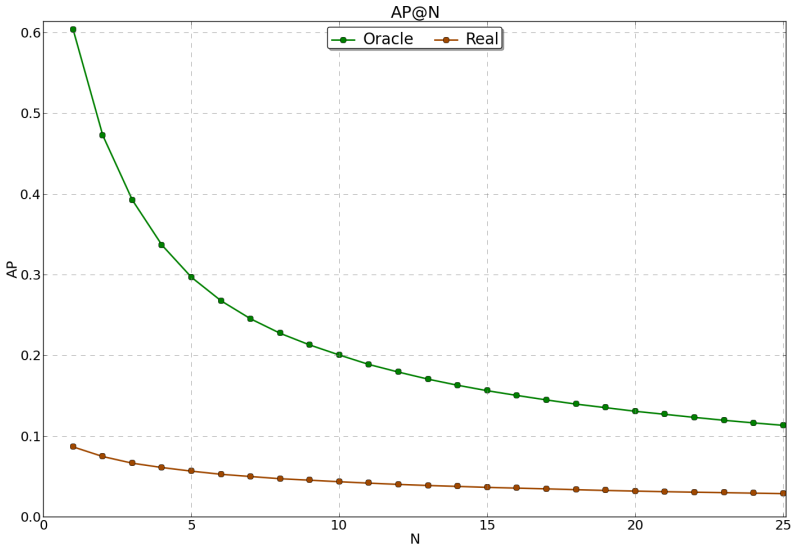


Figure 4.12.: The averaged Precision@N for $N \in \{1, \ldots, 25\}$ of the Concept Vocabulary based tag suggestion system. Comparing the real and the oracle system.

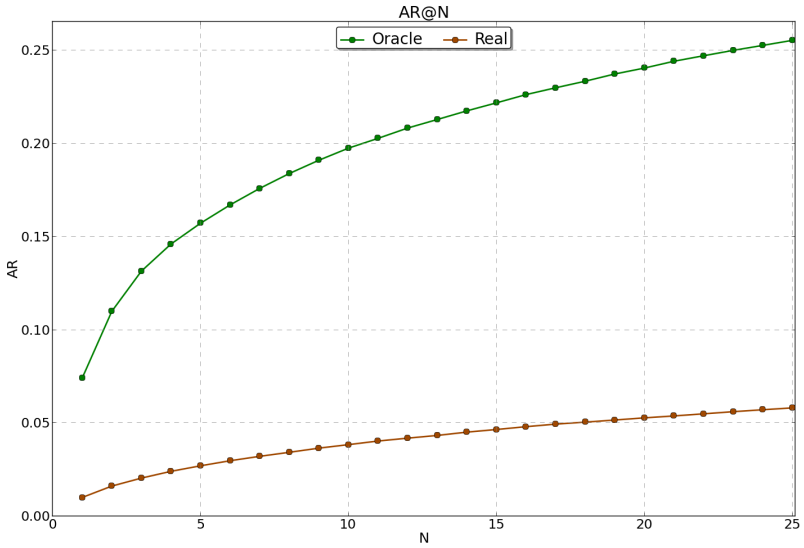


Figure 4.13.: The averaged Recall@N for $N \in \{1, \ldots, 25\}$ of the Concept Vocabulary based tag suggestion system. Comparing the real and the oracle system.

4.6.3. Nearest Neighbor Transfer Approach

As is shown in the previous two sections, the decision of the concept detection is not expressive enough to generate tags with high Precision. This supports the use of the Nearest Neighbor Transfer based system which uses the Nearest Neighbors for transferring tags, rather than the detected concept. This indeed performs better, as can be seen in Figure 4.15. The best averaged Precision value is reached for $N = 1$ and is about three percentage points above the Concept Vocabulary based approach. The best averaged Recall (for $N = 25$) is about 9%. With this, the system's performance is high above the Global lower bound and even considerably above the Concept Vocabulary approach, but performs worse than the Co-Occurrence based system. Examples can be seen in Figure 4.14. For the best performing of the examples, the average distance to its 50 Nearest Neighbors is about 0.8. The same measure has a value of 1.3 for the worse example.

U: piano, music, video, cat, nora, musical, pets, cats
S: nora, piano, cat, music, betsy, pets

U: luxor, amenhotep, egyptian+ruins, egypt+history
S: orphanage, krt, egypt, luxor, hip hop, misfits

Figure 4.14.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by the Nearest Neighbor Transfer based system. Precision@6 is from left to right: top values, bottom values. The middle image is omitted as no video near AP@6 exists in the test set.

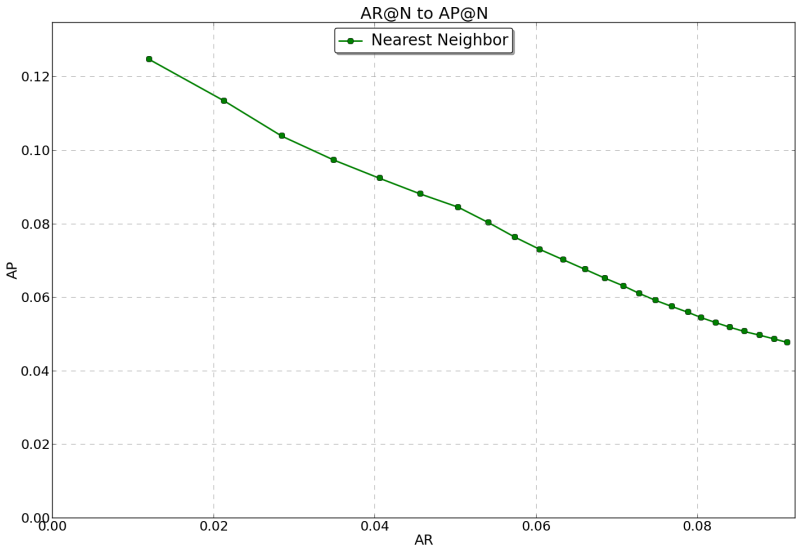


Figure 4.15.: The averaged Precision@N plotted against the averaged Recall@N, both of the Nearest Neighbor based tag suggestion system.

4.7. Visual Personalized Tag Transfer Fusion

The Visual Personalized Tag Transfer Fusion system depends on only one parameter, *perfrac*. This parameter reflects how strongly this system relies on the History (the higher the value, the bigger the History's influence). This parameter is a number between 0 (equal to the system described in Section 3.6.4) and 1 (similar to the History based system, but enhanced with visual re-ranking) and is denoted in brackets behind the system in figures (e.g. "PersonalizedTagTransfer(0.6)") if compared with other systems or just as *perfrac* if it is clear that the Visual Personalized Tag Transfer system is used (e.g. "$perfrac = 0.3$"). Figures 4.17 and 4.18 show the performance of this system for all *perfrac* between 0 and 1 with a step size of 0.1. A bar chart is used, as this makes comparing the values easier than with the representations used before. For better visibility only every second N for averaged Precision/Recall@N is shown. These figures indicate that, at least for this dataset, a *perfrac* between 0.6 and 0.8 is optimal. The maximum averaged Precision@N is achieved at $N = 1$ for *perfrac* = 0.7 with a value of about 47.5% and even for $N = 25$ the averaged Precision is still at about 15.6%. The maximum averaged Recall is as well achieved for *perfrac* = 0.7, but for $N = 25$, and with a value of nearly 32.1%. It can also be seen that both extreme cases for *perfrac* (0 and 1) perform considerably worse than the systems with the top values for *perfrac* and that their performance decreases faster. Three example results are found in Figure 4.16. All three results are for a *perfrac* of 0.7. The best performing example has a history that consists mainly of Michael Jordan commercials (8 of 12) and an average distance to its 50 Nearest Neighbors of 1.32, whereas the worst example's distance to its Nearest Neighbors is 1.34 and normally has a gardening and cooking centric tagging history. The two examples on the left and middle show that the Nearest Neighbors introduce tags to that systems that fit the content (`game`, `chemicals`) but are not considered correct, as they were not used by the uploading users.

Better results might be achieved, if *perfrac* would not be chosen globally, i.e. the same for all videos, but rather on a per video basis. The problems with this and some of the solutions to these problems will be shown in the context of the Weighted Sum based fusion in Section 4.8.2, as not only weights are considered there but an individual *perfrac* as well.

U: michael, jordan, nike, gatorade, commercial, air
S: michael, jordan, air, commercial, nike, game

U: valetpro, bilberry, safe, wheel, cleaner
S: sheeting, de, wheel, chemicals, cleaner, gel

U: free, videos, expertvillage, breakdancing, dancing, dance, dancelessons, break
S: cooking, food, recipe, home, diy, cake

Figure 4.16.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by the Visual Personalized Tag Transfer based system with *perfrac* = 0.7. Precision@6 is from left to right: top values, near AP@6, bottom values.

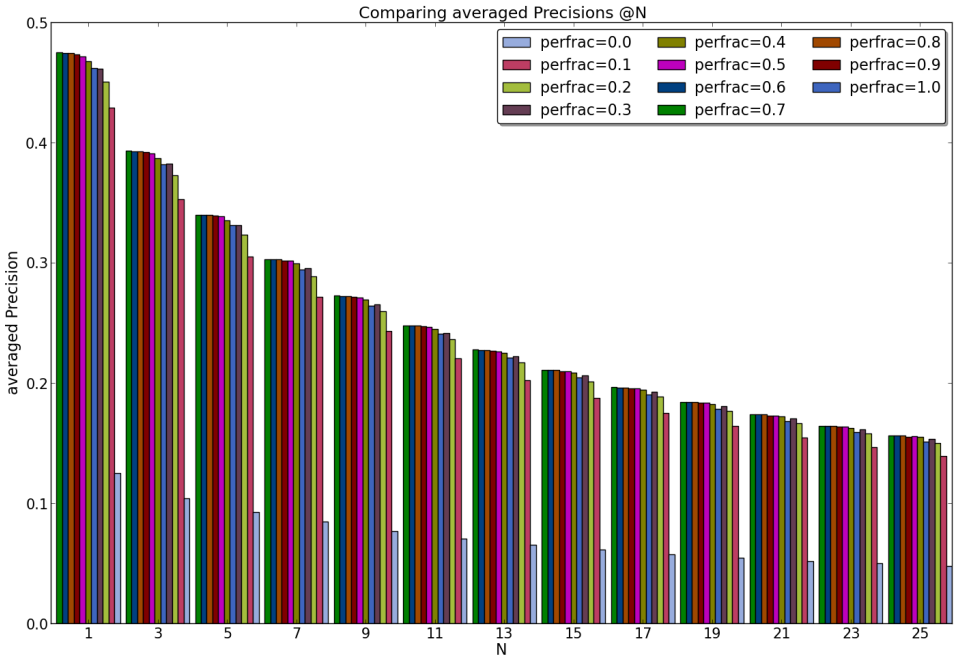


Figure 4.17.: The averaged Precision@N for $N \in \{1, 3, 5, \ldots, 25\}$ of the multiple Visual Personalized Tag Transfer based systems. The systems are denoted by their *perfrac*.

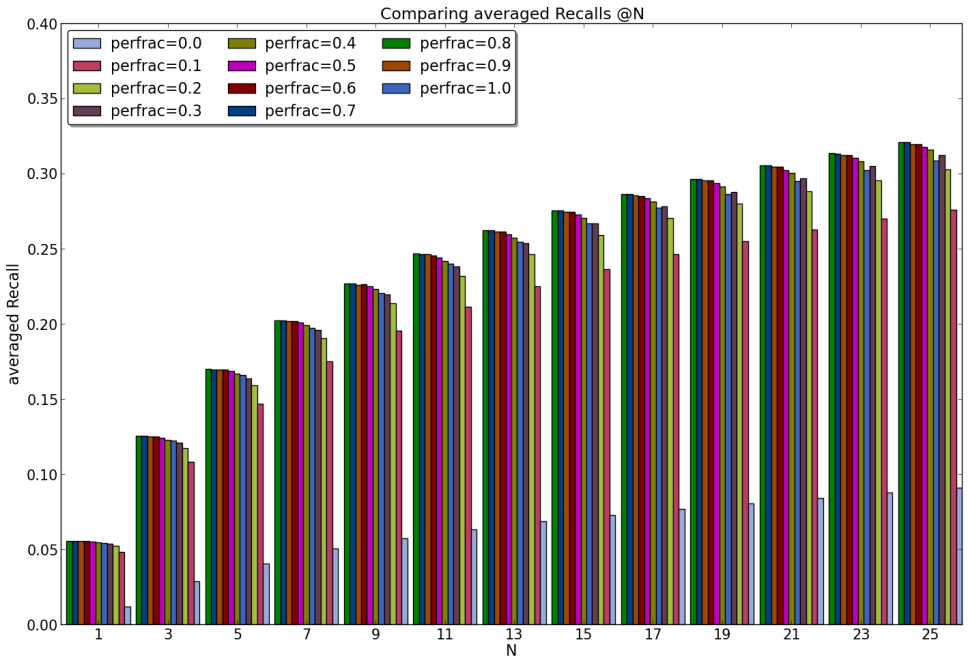


Figure 4.18.: The averaged Recall@N for $N \in \{1, 3, 5, \ldots, 25\}$ of the multiple Visual Personalized Tag Transfer based systems. The systems are denoted by their *perfrac*.

4.8. Fusion

In this section the multiple fusion approaches discussed in Section 3.8 will be evaluated. Furthermore, the finding of the parameters and, where applicable, the weights will be described in more detail and several choices will be compared to each other.

4.8.1. Rule Based Fusion

For the simple Rule based system, described in Section 3.8.1, two global parameters have to be tuned. This was done with a simple grid search[6]. It should be considered that the grid search was done on the whole V_{test} and therefore the performance of this system serves only as an upper bound, as the parameters might only fit this set of videos but might not generalize well, an effect that is called "overfitting". This effect might not be too strong though, because of the small number of parameters and their discrete nature, as well as the randomness of the data. Figure 4.19 and Figure 4.20 compare five of the best performing combinations in terms of averaged Precision and Recall respectively. Whenever one of the Rule based systems is shown in a figure, it will be denoted with a two digit representation hc. Here h is the minimum History length for a user to still rely on the History based system, denoted as *minhist* in Section 3.8.1, and c is the minimum number of Channels a user has to have subscribed to rely on the Channel based system, denoted as *minchan*. For example 12 would then mean a Rule based system with $minhist = 1$ and $minchan = 2$. As can be seen in the figures, systems with parameter combinations that strongly favor the History based tag suggestion system perform best. It can also be seen that the value for *minchan* does not have a great influence. The best performing system ($minhist = 1$ and $minchan = 1$) performs very similar to the History based system, as it uses a different approach for the slightly more than 1,000 users without history only. This, too, is the reason why it always performs at least on par with the History based system, but sometimes even surpasses it.

[6] See Section 3.8.1.

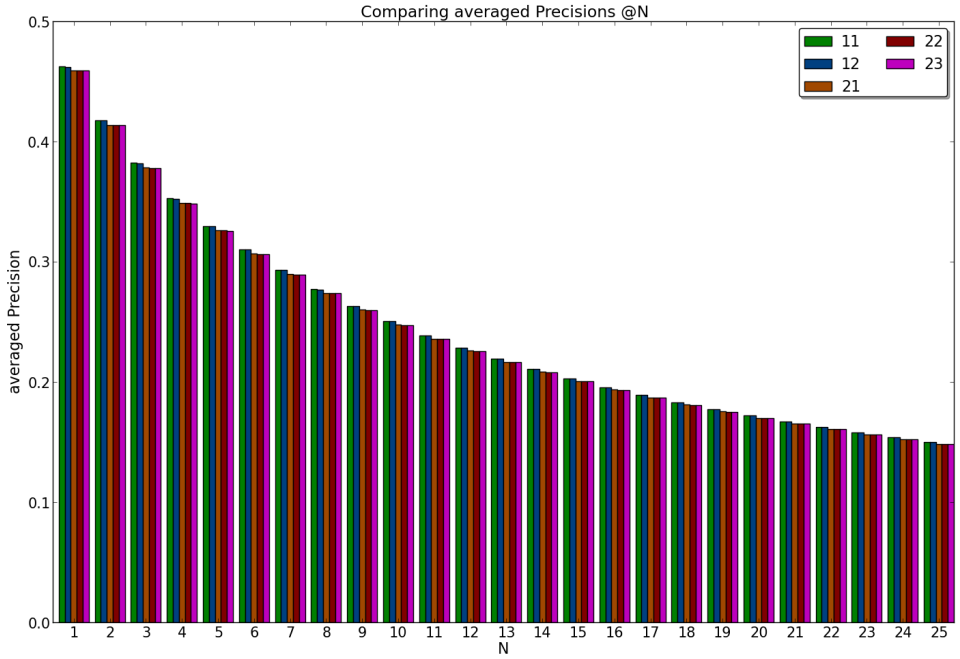


Figure 4.19.: The averaged Precision@N for $N \in \{1, \ldots, 25\}$ of the multiple Rule based systems. The systems are denoted as hc, with $h = minhist$ and $c = minchan$.

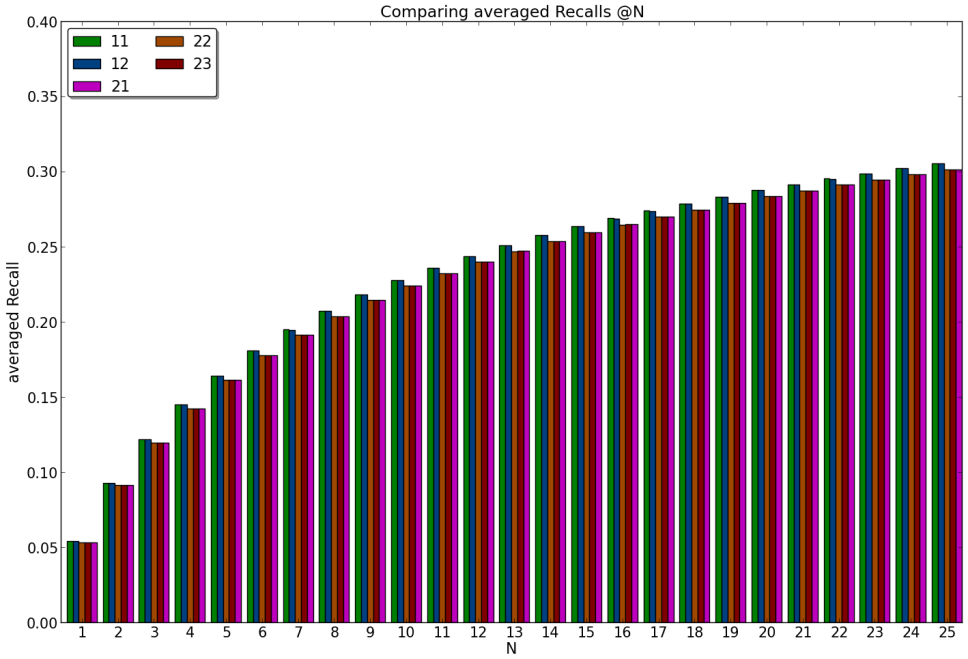


Figure 4.20.: The averaged Recall@N for $N \in \{1, \ldots, 25\}$ of the multiple Rule based systems. The systems are denoted as hc, with $h = minhist$ and $c = minchan$.

4.8.2. Weighted Sum Based Fusion

This section evaluates the Weighted Sum based fusion. As the weights and the finding of those weights is a crucial part of this system, three approaches to find weights are discussed before. For this implementation four tag suggestion systems were used for fusion: the History based approach (abbreviated with "History"), the Co-Occurrence based system (abbreviated with "Co-Occurrence"), the Channel based system (abbreviated with "Channel") and a visual system, namely the Visual Personalized Tag Transfer based approach (abbreviated with "PersonalizedTagTransfer"). With this, four weights and the *perfrac* parameter (for the Visual Personalized Tag Transfer system) have to be tuned.

4.8.3. Finding weights

As seen in Section 3.8.2, the Weighted Sum fusion strongly depends on the weights chosen. Therefore it is necessary to calculate the weights in a reasonable manner. The weights considered here are numbers in the range from zero to one and always sum up to one. This suffices, as all multiples of constellations have the same effect (e.g. $(0.1, 0.4, 0.5)$ results in the same list of suggested tags as $(1, 4, 5)$) and because only the relative relation between the weights is relevant (e.g. it is still possible to choose the weights in a way that all systems have the same influence, i.e. $(0.25, 0.25, 0.25, 0.25)$). Furthermore, the personalization fraction *perfrac* needed by the Tag Transfer system (see Section 3.7) is tuned in the same way the weights are calculated (but does not count to the sum restriction). In the following, three ways will be described to find suitable weights.

Oracle Weights

The first approach uses a grid search to determine specific weights for every target user which has to been considered an unrealistic upper bound for the Weighted Sum fusion, as it uses knowledge that is not available in a real setup. Every weight combination that fulfills the requirements (tested with a step size of 0.1), together with all possible values for *perfrac* (with a step size of 0.2), is tested and the ones with the highest Average Precision (Precision@N averaged over all $N \in \{1, \ldots, 25\}$) for the respective user are considered for this approach. If multiple combinations have the same Average Precision, all of them are stored and one of them is randomly picked for evaluation purposes. This is called the oracle version of this system, as it is able to predict the best weight combination for a given video.

Global Weights

The second approach tries to find one global weight combination that suits every video. This is again done with a grid search, although this time it is not quite as unrealistic, because the danger of overfitting is reduced by the large corpus of data and its random nature. Furthermore, the weights could be learned on a held out subset of V_{test} and then evaluated on the rest of V_{test}. For this again every weight combination, with all 11 values

for *perfrac* each, is tested, but this time the same combination is tested for every video and the ones with the best averaged Precision are considered.

Learned Weights

The third option is to determine users that are similar to the user $u_{v_{new}}$ and learn a suitable weight combination for the video of $u_{v_{new}}$ from their optimal weights. The optimal weights for other users can be gained as seen in the first approach. Learning the weights for $u_{v_{new}}$ is done in a leave-one-out fashion[7], meaning that the optimal weights generated by the first approach are considered known for every user, except the one that weights are learned for, denoted as u_{learn}. The set of users with known optimal weights is denoted as $U_{loo} = U_{test} \setminus u_{learn}$, with U_{test} the set of all users who correspond to the videos in V_{test}. Now the k Nearest Neighbors of u_{learn} in U_{loss} are calculated, using following features:

- the number of tags used in the user's history
- the average distance to the global Nearest Neighbors of the video associated with the user
- the average distance to the local Nearest Neighbors (as seen in Section 3.7) of the video associated with the user
- the sum of the number of videos in all subscribed channels
- the average interval between the uploading of the user's videos

Then the weighted average of the weights provided by these Nearest Neighbors is calculated, meaning that the weights of the closest Nearest Neighbor gets k votes for the average, whereas the farthest Nearest Neighbor's weights get only 1 vote. This weighted average is then taken as the weight combination for u_{learn} (except for the finding of weight combinations of other users, where its optimal weights are still considered). For an illustration of this approach to learning the weights, see Figure 4.21.

To find the features, the average values of several of the users' aspects were examined. Those were taken that showed a correlation with the performance of the subsystems (e.g. if the average number of tags in the history is lower, the History based tag suggestion system performs worse in average). As these features produce values from different ranges, they first have to be normalized. For this the Standard Score is used, meaning that every feature value x that is part of the set of feature values (one for each video) F is normalized in a way that the mean ($\mu_{normalized}$) of $F_{normalized}$ is 0 and the standard deviation ($\sigma_{normalized}$) is 1. This is achieved by using this formula: $x_{normalized} = \frac{x-\mu}{\sigma}$. A motivation for this formula can be found in Section 4.3 of [12].

[7]For more detail on this approach, also called the deleted estimate, see [7].

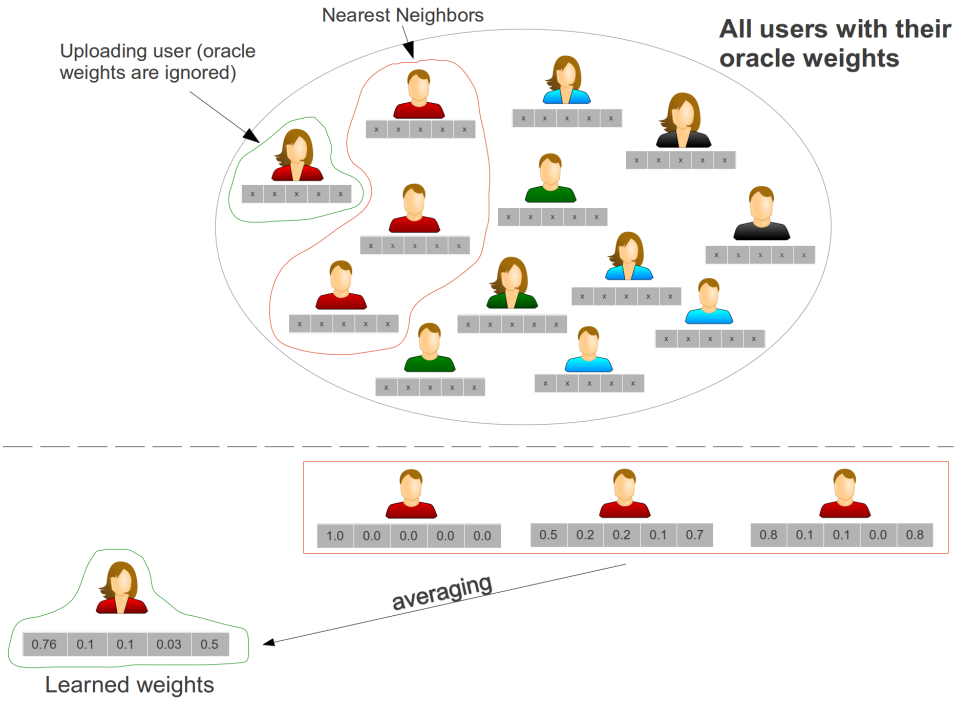


Figure 4.21.: The weights of a Weighted Sum based fusion are learned from the optimal weights of other users in a leave-one-out fashion.

4.8.4. Evaluation

In Section 3.8.2 it was discussed that either scores or ranks could be used for the weighted sum. As the score based version performed considerably worse in early stages of the research, the following will only discuss rank based versions.

Oracle Weights

First, the oracle system will be evaluated to establish an upper bound to which the realistic implementation can be compared. The performance of this system can be seen in Figure 4.23, comparing both averaged Recall and averaged Precision. This approach performs considerably better than the systems seen so far, with an averaged Precision of about 57.2% at $N = 1$ and it is still able to suggest tags with an averaged Precision of more than 30% for $N = 10$. Moreover, the maximum averaged Recall is nearly 37.2% for $N = 25$. Example results can be found in Figure 4.22. The best performing example has an average weight tuple of about (0.32, 0.3, 0.13, 0.26) and the Visual Personalized Tag Transfer's *perfrac* is 0.6, which means that in the average of all best performing weight combinations the History based system has a weight of 0.32, the Visual Personalized Tag Transfer(0.6) has a weight of 0.3, the Channel based system gets a weight of 0.13 and the Co-Occurrence based system has a weight of 0.26. For the example that performs near

the averaged Precision, this weight tuple is (0.25, 0.1, 0.4, 0.25) with *perfrac* = 0.6. If the best value for Precision is 0 (as it is for the worst example), all weight combinations are equally unsuited for this video and therefore the average weight tuple carries no information.

Another interesting information are the weights which were chosen by this approach in general. This can be seen in Figure 4.24. If for a given video several weight combinations performed the same, all of these combinations are counted. If the Precision for a given video was zero for the fusion, these weights are not counted, as all combinations perform the same in this case. How many videos fall in this category can be seen in the same figure. What can be seen there is that rarely only a single system is relied on (the higher weights are the more seldom ones) and that the History and Visual Personalized Tag Transfer based systems get higher weights in general and are far less often completely left out (a weight of 0). What can also be seen is that the different values for *perfrac* are quite evenly distributed except for *perfrac* = 0 which occurs less often.

U: suzuki, gsxr1300, gsxr, 1300, hayabusa, smcbikes, sheffield, motorcycle, motorbike, bike
S: motorcycle, motorbike, sheffield, bike, smcbikes, ducati

U: snow, uk, funny, east-midlands, east, midlands, leicester, first, bus
S: leicester, pakistan, syria, uk, imran, bbc

U: concrete, compressive, strength, testing, machine
S: police, kingdom, emergency, gymnastics, show, travel

Figure 4.22.: Each image is a keyframe representing a video on YouTube. Under each keyframe are the video's tags and the ones suggested by using the Weighted Sum based fusion using oracle weights. Precision@6 is from left to right: top values, near AP@6, bottom values.

Global Weights

For this system the weight combination of $(1, 0, 0, 0)$ has shown to be the best. This means that using only the History based system provides the best performance in terms of averaged Precision. This shows that using a single global weight combination for all videos is not a good idea, implying the need for personalizing the weights. As this system performs exactly like the History based tag suggestion system which is evaluated in Section 4.3, this system is not evaluated again.

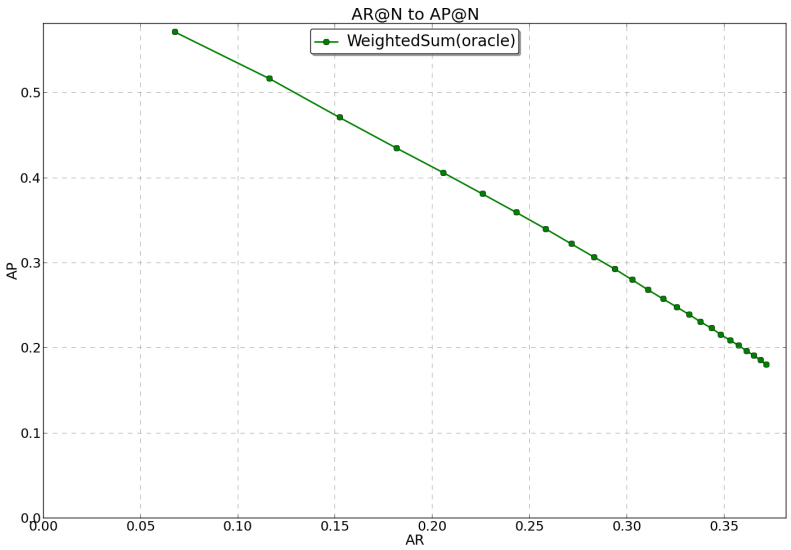


Figure 4.23.: The averaged Precision@N plotted against the averaged Recall@N, both of the Weighted Sum based tag suggestion system, using oracle weights.

Learned Weights

As this system averages the weights gained from the Nearest Neighbors, it is of interest how averaging affects the performance of the Weighted Sum based system. Some insights can be gained from Figure 4.25, in which the oracle system is compared to a system for which not one of the equally performing oracle weights is randomly picked, but rather the average of those is calculated. As can be seen there, calculating the average has a tolerable influence on the performance.

The system as it is evaluated here uses all features described in Section 4.8.3 and learns weights from $k = 5$ Nearest Neighbors. Its performance compared to the oracle system can be seen in Figure 4.26 for averaged Precision and in Figure 4.27 for averaged Recall. It can be seen there that the real system does not reach the upper bound provided by the Oracle system. It still has an averaged Precision nearly 46% for $N = 1$ and an averaged Precision of about 15.2% for $N = 25$ but especially the values in between ($N = 3, \ldots, 12$) compare unfavorably to the History based system. The same can be observed for the averaged Recall although a bit less distinct. This might be due to the features which might not be descriptive enough or just too few, or to the Nearest Neighbor approach to learning – similar users might not need similar weights.

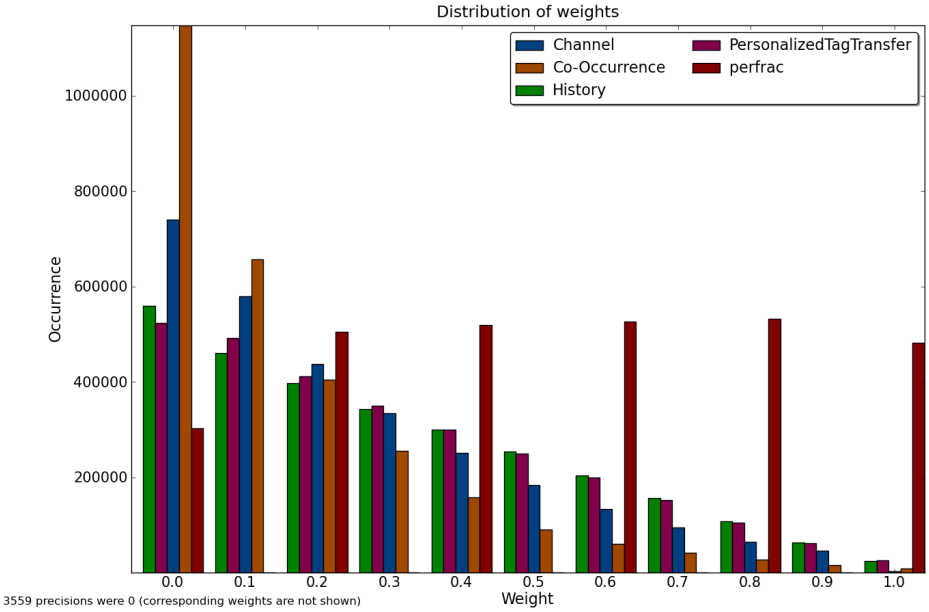


Figure 4.24.: The distribution of weights as chosen by the oracle Weighted Sum fusion approach. The number of videos with Precision=0 is given, as these are not shown in the statistic.

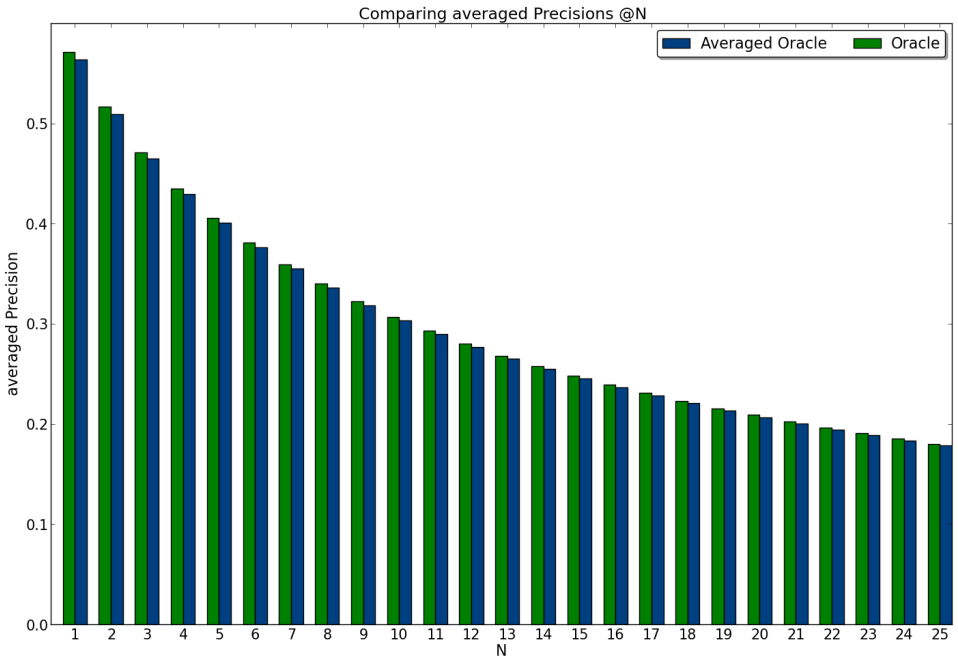


Figure 4.25.: Comparing the Oracle Weighted Sum approach (Oracle) and the Weighted Sum with averaged oracle weights (averaged Oracle) in terms of averaged Precision.

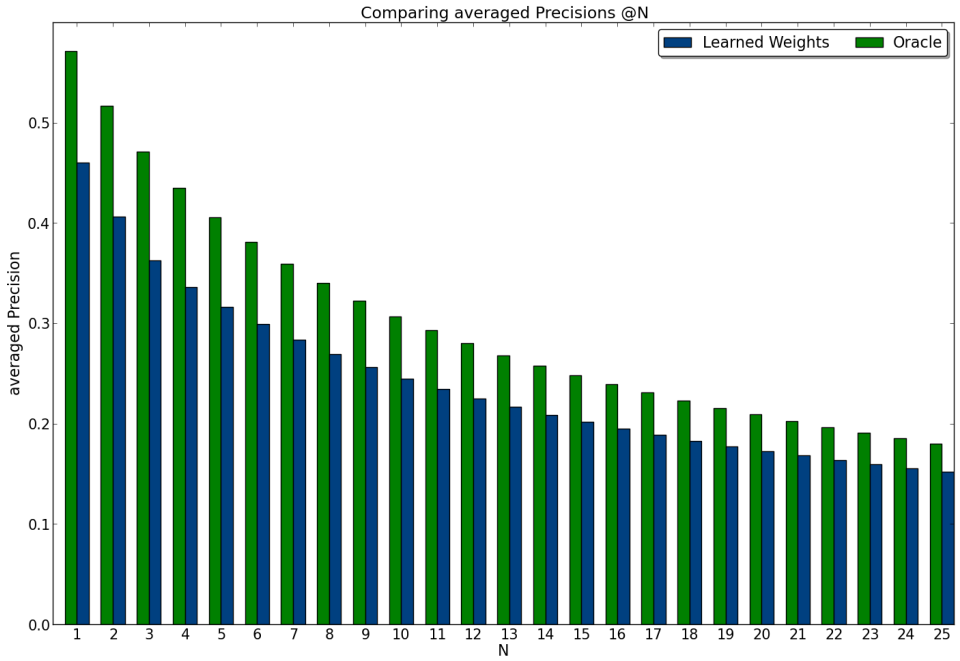


Figure 4.26.: Comparing the Oracle Weighted Sum approach (Oracle) and the Weighted Sum with learned weights for $k = 5$ (Learned Weights) in terms of averaged Precision.

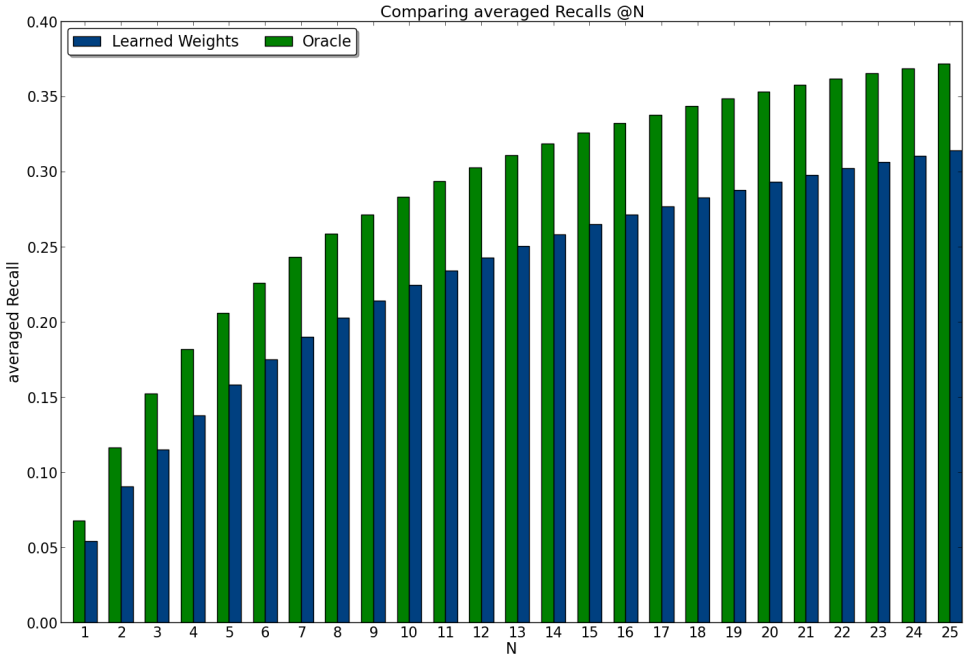


Figure 4.27.: Comparing the Oracle Weighted Sum approach (Oracle) and the Weighted Sum with learned weights for $k = 5$ (Learned Weights) in terms of averaged Recall.

4.9. Comparison

This section will give a comparative analysis of the systems described in the previous chapter. As the Global Tag Statistic based system was introduced as a sensible lower bound for the performance of a tag suggestion system, the single tag suggestion systems will be compared to this. And as the History based system is a system that has proven itself in several real setups, the fusions of the systems will be compared to this. For ease of notation, all systems have abbreviated descriptions that will be used in the legends for the coming figures. An overview of all abbreviations can be seen in Table 4.2.

Table 4.2.: An overview over the systems and their abbreviations.

System	**Abbreviation**
Global Tag Statistic Based System	Global
History Based System	History
Co-Occurrence Based System	Co-Occurrence
Channel Based System on all/those with channel	Channel(all/wc)
Oracle/Real Concept Vocabulary Based System	Vocabulary(oracle/real)
Nearest Neighbor Transfer Based System	Nearest Neighbor
Rules Based Fusion, $minhist = x$, $minchan = y$	Rule(xy)
Visual Personalized Tag Transfer Based Fusion, $perfrac = \alpha$	PersonalizedTagTransfer(α)
Oracle/Learned Weighted Sum Based Fusion	WeightedSum(oracle/learned)

Single Systems

In Figure 4.28 and Figure 4.29 the single systems' averaged Precision@N and averaged Recall@N are compared to each other. For better visibility, only every second N is plotted. It can be seen that there is indeed no single system that is able to clearly outperform the History based system, nor is there a system performing worse than the Global Tag Statistic based system. The Oracle Concept Vocabulary based system outperforms the History based system for $N = 1$ considerably, but looses in performance much faster than the History based system. This might be due to the fact that the concept vocabularies are too noisy and only the really widely used tags are of meaning for a lot of users uploading videos with this concept present. The Real Concept Vocabulary based system performs much worse which was to be expected given the performance of the concept detection pipeline. The optimistic Channel based approach outperforms the Co-Occurrence based system for $N = 1$ but its performance decreases much faster. The Nearest Neighbor Transfer based approach performs quite good, especially when compared to the real Concept Vocabulary based system, but it, too, is far from surpassing the History based system. For a detailed overview of the performances see Table A.4 and Table A.5.

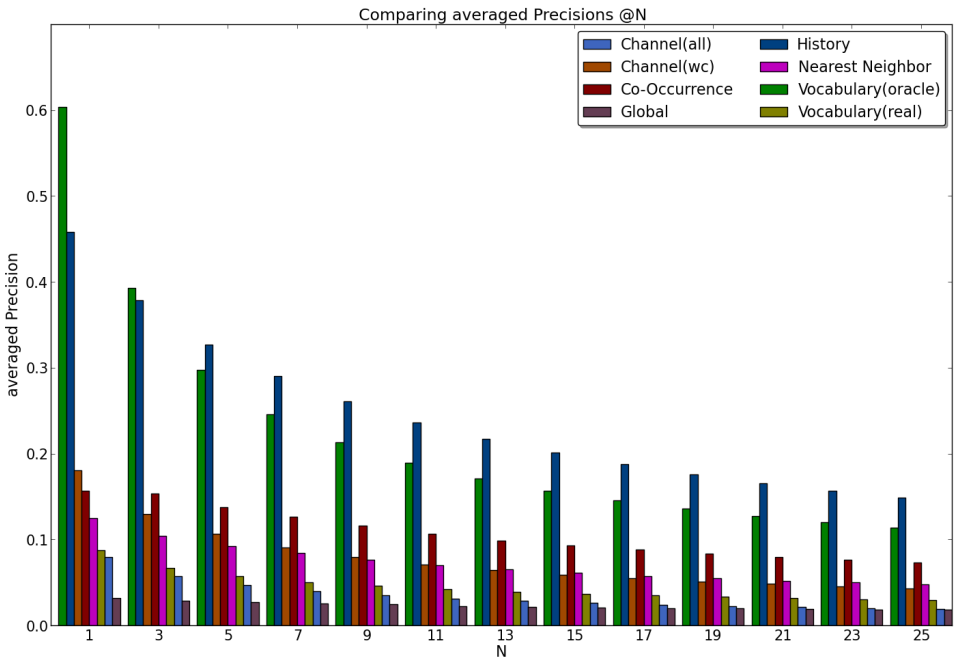


Figure 4.28.: The averaged Precision@N for $N \in \{1, 3, 5, \ldots, 25\}$ of all single systems.

Fusion Systems

For the fused systems the performance comparison can be seen in Figure 4.30 for averaged Precision and in Figure 4.31 for averaged Recall. It can be seen that the Oracle Weighted Sum based fusion considerably outperforms the History based system. The Visual Personalized Tag Transfer based system, too, outperforms the History based system, especially for smaller N. Even the simple Rule based system is able to outperform the History based system, although not by much and is often only on par with it. The Weighted Sum fusion that uses learned weights does not reach its upper bound, in fact it sometimes even performs worse than the History based system. For these systems, too, a more detailed overview of the performances can be found in the Appendix, in Table A.6 and Table A.7.

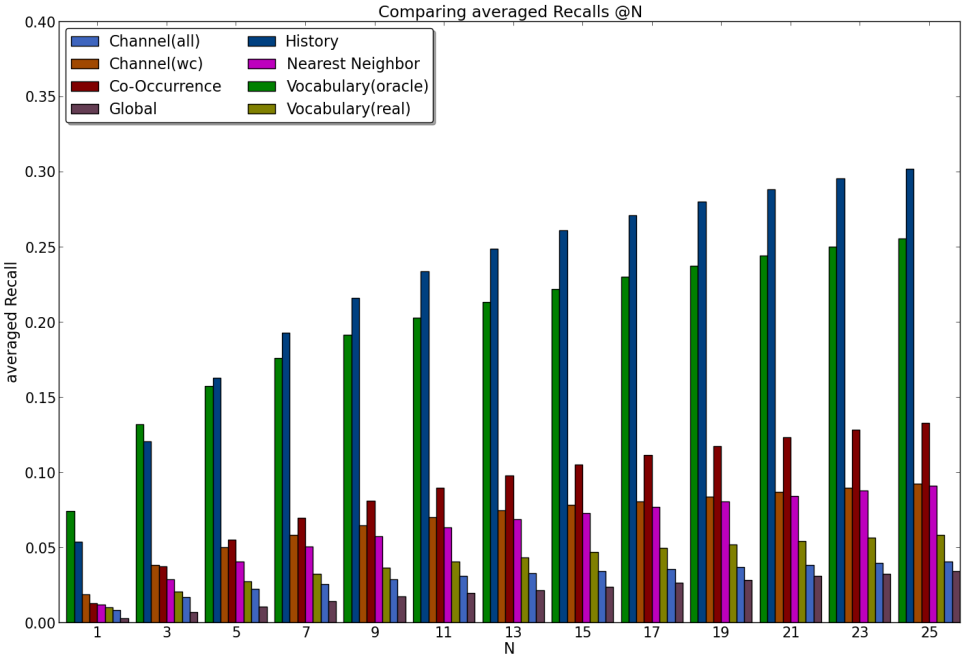


Figure 4.29.: The averaged Recall@N for $N \in \{1, 3, 5, \ldots, 25\}$ of all single systems.

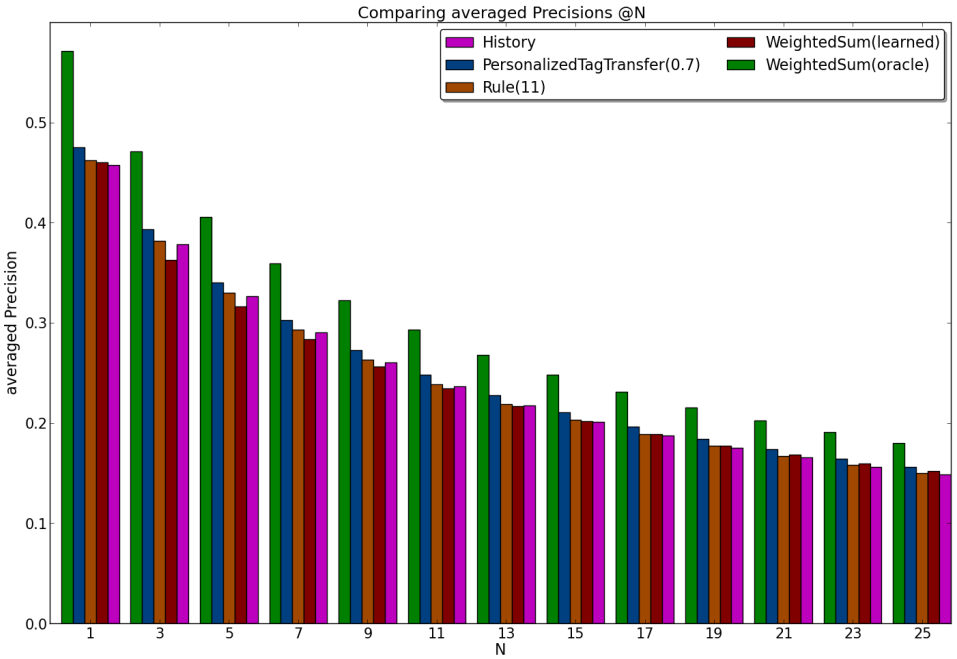


Figure 4.30.: The averaged Precision@N for $N \in \{1, 3, 5, \ldots, 25\}$ of all fused systems.

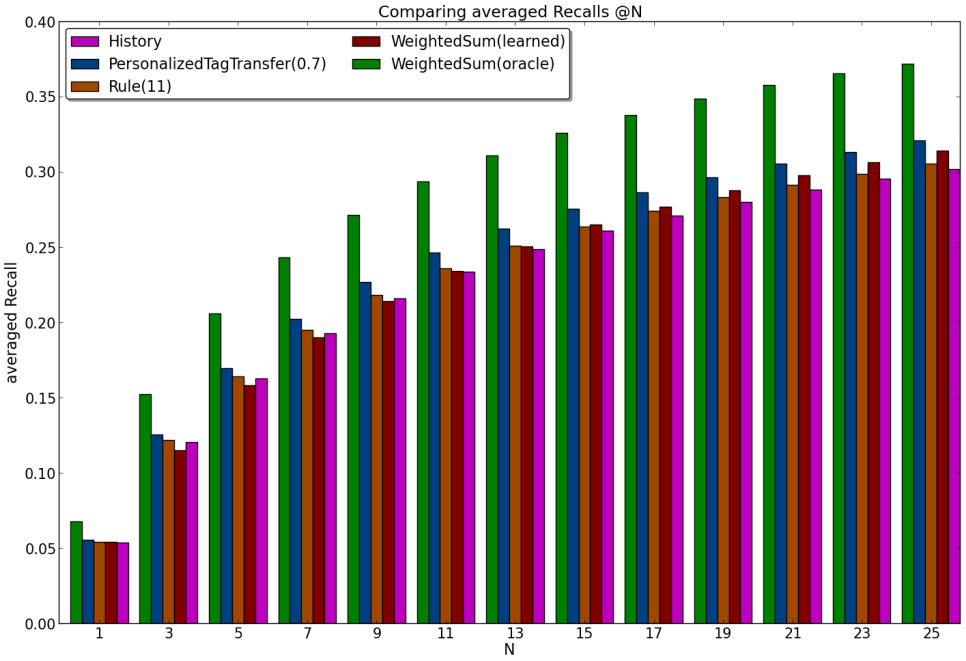


Figure 4.31.: The averaged Recall@N for $N \in \{1, 3, 5, \ldots, 25\}$ of all fused systems.

5. Conclusion and Outlook

This chapter will discuss the results presented in Chapter 4 and show to what extent this thesis' goals were achieved. Furthermore, an outlook will be given on what might be sensible future improvements to the systems presented here.

5.1. Conclusion

As has been discussed in Section 1, millions of people handle a vast amount of video on a daily basis and with this the need for tagging is ever higher. To support the user in his tagging activity, this thesis seeks to provide tag suggestion systems which are able to suggest tags the user him-/herself would use. For this a comprehensive overview of possible approaches was given, as well as multiple approaches to fuse those systems into a more powerful one which is able to suggest tags itself. Each of the systems was described in a manner that makes it easy to implement. Furthermore, every system was evaluated and compared to the other systems such that it is possible to see which systems perform better than the others, also giving a good starting point for solving similar problems. This overview showed the merits of the different modalities and their combinations and also discussed possible reasons for the different performances.

The main novelty of this thesis was a comparative study of systems that incorporate information from a high number of modalities and unimodal systems. The modalities included social signals (both general and personalized) and content signals, as well as the information provided by the user him-/herself. Several approaches to get a multimodal system by fusion of single modal systems were proposed, thus creating systems that surpass the purely History based systems and illustrating, by means of an oracle system, that there is still even more potential for this kind of approach. In addition to describing these fusion systems, this thesis also presented possible ways of learning parameters and weights needed by these systems.

All this resulted in two especially noteworthy systems. The first being the Visual Personalized Tag Transfer based tag suggestion system, which merges global Nearest Neighbors with the visually re-ranked History of the user to suggest tags. This system incorporates visual information and information about the user and is able to surpass the History based system's performance, while only needing one, easy to tune, parameter. This system can be used on a real setup and can be used with different features, e.g. color histograms, and, with suitable features, is also applicable to most other domains, e.g. the image domain.

The second is the Weighted Sum based fusion which performs even better than the Personalized Tag Transfer system for oracle weights, showing the great potential of this

approach, although this potential could not be fully reached in the scope of this thesis. Furthermore, this system is easily extensible to incorporate even more than the four modalities that are proposed here, while still needing only one parameter per modality.

5.1.1. Discussion

The experiments have shown that indeed the History based system clearly outperforms all single modal systems, although the oracle version of the Concept Vocabulary based system suggests that, for a very good description of the visual contents, systems based on visual signals alone might be able to surpass the History based approach. They have also shown that all systems perform better than using a global tag statistic and that at least three of the proposed single modal systems perform acceptable, namely the Channel based, the Nearest Neighbor Transfer based and the Co-Occurrence based system.

Even the very simple fusion by a static rule is able to perform better than the History based system, even though only a little. Most noticeably, the Visual Personalized Tag Transfer approach that combines a user's history with the information gained by visual Nearest Neighbors of the video, is able to outperform the History based system, although only depending on a single global parameter. Especially high potential can be seen in the Weighted Sum fusion that fuses several systems into one. Its oracle version considerably outperforms every other system presented in this thesis, although it is also shown that finding these oracle weights is not easily done and that the performance of this system suffers harshly from wrong weights.

In this thesis the tag suggestion systems are evaluated on the original uploader's tags only, which means that additional fitting tags are not recognized as correct. Therefore, it should be considered that some of the systems might fare better if evaluated by less strict means (e.g. user studies). This can be seen in Figure 4.16, in which both the leftmost and the middle video would have gotten better values for Precision than they did in the actual evaluation if the additional fitting tags (`game`, `chemicals`) would be counted as correct.

5.1.2. Future Work

As fusion depends on the quality of the single tag suggestion systems, one way to improve the performance might be to provide more such systems, for example a Friends based system or a system that uses comments to find users interested in the same topics (a way to implement these two systems is already shown in Section 3.5). Another way to get better performance would be to improve the existing systems, this is especially true for the concept detection based systems, as these might greatly benefit from a better concept detection pipeline, for example one that is better suited for videos or uses SVMs[1] instead of Nearest Neighbor classification, which are currently regarded state-of-the-art in concept detection.

[1]For an overview of the functionality of SVMs and a discussion on their merits see [3].

The finding of weights for the Weighted Sum based fusion was a crucial part for this system. As this system's oracle version greatly outperforms all other systems, it might be of interest to enhance the learning of these weights. This could be done for example by finding more expressive features for the users (a value that expresses the coherence of the tag history for example might be of interest) or by utilizing a different approach than k Nearest Neighbors altogether. Another interesting possibility might be to use a more complex and potentially more powerful approach to fusion than the weighted sum, for example a RankBoost[2] based approach.

[2]See [9] for information on the RankBoost algorithm.

6. Bibliography

[1] Alexa. Alex Traffic Rank (Global Top Sites for 1 Month Statistic). `http://www.alexa.com/topsites/global`, 2012. Retrieved April 22, 2012.

[2] Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 971–980 (ACM, New York, NY, USA), 2007.

[3] Kristin P. Bennett and Colin Campbell. Support Vector Machines: Hype or Hallelujah? In *SIGKDD Explor. Newsl.*, 2(2):1–13, December 2000.

[4] Jean-Charles de Borda. Mémoire sur les élections au scrutin, 1784. URL `http://asklepios.chez.com/XIX/borda.htm`. After the version by Dr. Lucien de Luca, retrieved April 29, 2012.

[5] Denis Bouyssou, Thierry Marchant, Marc Pirlot, Patrice Perny, Alexis Tsoukiàs and Philippe Vincke. Evaluation and Decision models: A critical perspective. In *Kluwer's International Series* (Kluwer), 2000.

[6] Bertrand Clarke and Dongchu Sun. Reference priors under the chi-squared distance. In *Sankhyā. Series A. Methods and Techniques*, 59(2):215–231, 1997.

[7] Luc Devroye, László Györfi and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition* (Springer), 1996.

[8] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis* (John Willey & Sons, New York), 1973.

[9] Yoav Freund, Raj Iyer, Robert E. Schapire and Yoram Singer. An efficient boosting algorithm for combining preferences. In *The Journal of Machine Learning Research*, pages 933–969, 2003.

[10] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37 (Springer-Verlag, London, UK, UK), 1995.

[11] Markus Koch. *Ad Targeting for Web Video by Automatic Video Annotation*. Master's thesis, 2011.

[12] Richard J. Larsen and Morris L. Marx. *An Introduction to Mathematical Statistics and Its Applications (3rd Edition)* (Prentice Hall), 2000.

[13] Jia Li and James Z. Wang. Real-time computerized annotation of pictures. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 911–920 (ACM, New York, NY, USA), 2006.

[14] Xirong Li, Efstratios Gavves, Cees G.M. Snoek, Marcel Worring and Arnold W.M. Smeulders. Personalizing automated image annotation using cross-entropy. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 233–242 (ACM, New York, NY, USA), 2011.

[15] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150– (IEEE Computer Society, Washington, DC, USA), 1999.

[16] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297 (University of California Press), 1967.

[17] Jane Grossman Michael Grossman, Robert Katz. *The First Systems Of Weighted Differential And Integral Calculus* (Michael Grossman), 2006. ISBN 0977117014.

[18] Adam Rae, Börkur Sigurbjörnsson and Roelof van Zwol. Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 92–99 (Le Centre De Hautes Etudes Internationales d'Informatique Documentaire, Paris, France), 2010.

[19] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning (Information Science and Statistics)* (Springer), 2004.

[20] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *Readings in information retrieval*, pages 323–328 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA), 1997.

[21] Neela Sawant, Ritendra Datta, Jia Li and James Z. Wang. Quest for relevant tags using local interaction networks and visual content. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 231–240 (ACM, New York, NY, USA), 2010.

[22] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 259–266 (ACM, New York, NY, USA), 2008.

[23] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 327–336 (ACM, New York, NY, USA), 2008.

[24] Sanjay C. Sood, Sara H. Owsley, Kristian J. Hammond and Larry Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *International Conference on Weblogs and Social*. 2007.

[25] Ronald A. Thisted. *Elements of statistical computing: numerical computation* (Chapman & Hall, Ltd., London, UK, UK), 1988.

[26] George Toderici, Hrishikesh Aradhye, Marius Pasca, Luciano Sbaiz and Jay Yagnik. Finding meaning on YouTube: Tag recommendation and category discovery. In *CVPR*, pages 3447–3454. 2010.

[27] Lei Wu, Linjun Yang, Nenghai Yu and Xian-Sheng Hua. Learning to tag. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 361–370 (ACM, New York, NY, USA), 2009.

[28] YouTube. YouTube Most Viewed Of All Time (Charts). `http://www.youtube.com/charts/videos_views?t=a`, 2012. Retrieved April 22, 2012.

[29] YouTube. YouTube Press Statistics. `http://www.youtube.com/t/press_statistics`, 2012. Retrieved April 22, 2012.

[30] Mu Zhu. Recall, Precision and Average Precision. Technical report, Department of Statistics & Actuarial Science, University of Waterloo, 2004.

Appendix

A.1. Concepts

Table A.1.: A list of all semantic concepts used and the corresponding YouTube queries. This table is based on the original table A.1 in [11], kindly provided by Markus Koch.

Concept	Query	Category
airplane-flying	airplane & flying -indoor	-
americas-got-talent	americas got talent	-
anime	anime mix	-
aquariums	aquarium fish tank	Animals
arcade	arcade	Travel
asians	asians -hot -sexy -bikini	People
autumn	autumn colors	Travel
baby	baby first	People
badlands	badlands	Travel
balloons	balloons	Entertainm.
baseball	baseball -golf	Sports
basketball	basketball	Sports
beach	beach	Travel
beehive	beehive	Animals
bicycle	bicycle	Vehicles
bikini	bikini	
bill-clinton	bill clinton	News
birds	birds	Animals
blacksmithing	blacksmith	Howto
boat	boat small -rc	Vehicles
boat-ship	ship &(queen|freedom|royal)	Vehicles
boobs	boobs tits	
boxing	boxing	Sports
breakdancing	break dancing	
bridge	bridge -crossing -ship	Travel
brown-bear	brown bear	Animals
bus	bus -van -suv -vw -ride	Vehicles
cake	cake	Howto
camels	camel|dromedar -spider	Animals
campus	university campus tour	-
car	car	Vehicles
car-crash	car crash	Vehicles
car-racing	car racing -rc	Sports
cartoon	cartoon	Film
castle	castle &(afar|outside) -inside	Travel
cathedral	cathedral	Travel
cats	cats	Animals
celebration	celebration	Travel
cheerleading	cheerleading	-
choir	choir	-
christmas-tree	christmas tree -fire	-
circus	circus show	-
city-skyline	skyline	Travel
cityscape	cityscape -slideshow -emakina	Travel
classroom	classroom & school -secret	-

Continued on next page

Table A.1.: (Continued) A list of all semantic concepts used and the corresponding YouTube queries. This table is based on the original table A.1 in [11], kindly provided by Markus Koch.

Concept	**Query**	**Category**
clock-tower	clock tower	Travel-
clouds	clouds & beautiful	Travel
cockpit	cockpit -railway -line	Vehicles
commercial	commercial -barack	-
concert	concert	Music
cooking	cooking	Howto
counterstrike-game	counterstrike movie -lego -real	
court	court judge	News
cows	cow	Animals
crane	crane	Vehicles
crash	crash	Vehicles
dam	dam	Travel
dancing	dancing	People
dark-skinned-people	black people	-
darth-vader	darth vader	-
demonstration	protesting	-
desert	desert	Travel
dog	dog	Animals
dogs	dogs	Animals
drawing	drawing	Film
drinking	drinking competition	-
driver	car & vehicle & driver -simulator	-
drummer	drummer	Howto
eiffeltower	eiffeltower	Travel
emergency-vehicle	emergency & vehicle -driver -ride	Vehicles
excavation	excavation	Travel
explosion	explosion	Howto
fence	fence	Travel
fencing	fencing	Sports
ferarri	ferarri	Vehicles
firefighter	firefighter training	-
fireworks	fireworks (nice or beautiful)	-
fish	fish	Animals
fishing	fishing	Sports
flood	flood water	News
flower	flower & (bouquet|bloom)	-
food	food delicious	-
football	american football -soccer	Sports
forest	forest	Travel
fountain	fountain	Travel
freeclimbing	freeclimbing	Sports
furniture	furniture	-
garden	garden beautiful -royal -coral	Travel
gardening	gardening	Howto
gas-station	gas station	Travel
georgewbush	george w bush	News
geyser	geyser	Travel
glacier	glacier	Travel

Continued on next page

Table A.1.: (Continued) A list of all semantic concepts used and the corresponding YouTube queries. This table is based on the original table A.1 in [11], kindly provided by Markus Koch.

Concept	**Query**	**Category**
glasses	glasses wearing -not -are	-
golf	golf	Sports
golf-course	golf course flyover	Sports
graffiti	graffiti	-
grand-canyon	grand canyon	Travel
gym	gym	Sports
gymnastics	gymnastics	Sports
hand	hand & daft	-
harbor	harbor & dock	Travel
helicopter	helicopter	Vehicles
highway	highway us route	-
hiking	hiking	Travel
horse	horse	Animals
horse-racing	horse racing	Sports
hospital	hospital & emergency	-
hotel-room	"hotel room"	Travel
house	house sightseeing	Travel
ice-skating	ice skating	Sports
interview	interview	News
iphone	iphone	-
jewellery	jewellery	-
jungle	jungle tropical	Travel
kiss	kissing two	-
kitchen	kitchen -knife -remodel	Howto
laboratory	laboratory tour	-
laundry	laundry	Howto
lava	lava flow	Travel
library	library tour	-
lighthouse	lighthouse	Travel
lightning	lighting strike	Travel
map	map geographic	-
marionette	marionette show	-
market	market	Travel
mccain	john mc cain	News
memorial	memorial -day	Travel
military-parade	military parade	-
monitor	screen monitor	-
moon	moon footage	-
mosque	mosque	Travel
motorcycle	(motorcycle or motorbike) -crash	Vehicles
mountain	mountain & panorama	Travel
muppets	muppet show	-
music-video	music video	-
native-american	native american dance	-
neon-sign	neon sign	Travel
nighttime	"by night"	Travel
obama	barrack obama	News
office	office working	-

Continued on next page

Table A.1.: (Continued) A list of all semantic concepts used and the corresponding YouTube queries. This table is based on the original table A.1 in [11], kindly provided by Markus Koch.

Concept	**Query**	**Category**
old-people	"old people"	-
operating-room	operating room	-
orchestra	orchestra symphony	-
origami	origami	Howto
outer-space	universe galaxy -super -song	-
pagoda	pagoda	Travel
parachute	parachute -no	Sports
penguin	penguin	Animals
phone	phone & device	-
piano	piano playing	-
pier	pier	Travel
playground	playground	Travel
poker	poker	Entertainm.
polar-bear	polar bear	Animals
pope	pope benedict	-
pottery	pottery	-
press-conference	press conference	News
procession	procession	Travel
pyramids	pyramid	Travel
race	race	Vehicles
railroad	railroad train -model	Vehicles
rainbow	rainbow beautiful	Travel
rainforest	rain forest	Travel
ranch	ranch	Travel
rc-car	rc car	Vehicles
restaurant	restaurant	Travel
rice-terrace	rice terrace	Travel
riding	horse riding	-
riot	riot	News
river	river	Travel
robot	robot -dance -dancers	-
rocket-launching	rocket launch -model -mini -toy	-
rodeo	rodeo bull riding	Sports
rooftop	rooftop	Travel
rugby	rugby	Sports
ruins	ruins -underwater	Travel
runway	runway airport	-
safari	safari	Travel
sailing	sailing	Travel
santa	santa (costume or outfit)	-
secondlife	secondlife	Games
shipwreck	ship wreck	Travel
shooting	shooting gun	-
shopping-mall	shopping (mall or center)	Travel
simpsons	the simpsons homer	-
singing	singing & (gospel|choire)	-
skateboarding	skateboarding	-
skiing	skiing -water	Sports

Continued on next page

Table A.1.: (Continued) A list of all semantic concepts used and the corresponding YouTube queries. This table is based on the original table A.1 in [11], kindly provided by Markus Koch.

Concept	**Query**	**Category**
sky	beautiful sky	Travel
snake	snake	Animals
snooker	snooker	Sports
soccer	soccer	Sports
soldiers	soldiers -child	News
stairs	stairs	Travel
steppe	steppe	Travel
street	street & paved	-
submarine	submarine	Vehicles
subway	subway station	Travel
sunrise	sunrise	Travel
surfing	surfing wave	-
swimming	swimming	Sports
swimming-pools	swimming pool	Travel
sword-fight	sword fight	Sports
talkshow	talkshow	People
tank	tank	Vehicles
tennis	tennis -table	Sports
tent	tent	Travel
themepark	park & (amusement|theme)	Travel
toilet	toilet	-
tony-blair	tony blair	News
tornado	tornado	-
tractor-combine	(harvester or tractor)	Vehicles
traffic	traffic	Travel
traffic-lights	traffic lights	Travel
tunnel	tunnel & (through|inside) -approach	Travel
turban	turban	-
two-people	two & people -sleepy -questions	-
underwater	underwater	Travel
us-flag	US flag raised	-
vending-machine	vending machine	Travel
videoblog	videoblog	People
waterfall	waterfall	Travel
weather	weather forecast	-
wedding	wedding footage	-
wheel	wheel	Vehicles
windmill	wind mill	Travel
windows-desktop	windows desktop	-
worldofwarcraft	world of warcraft	Entertainm.
wrestling	wrestling	Sports

A.2. Stop Words

Table A.2.: The list of Stop Words used for cleaning the vocabulary. The reduction is done case-insensitive.

a	about	above	across	after	again
against	all	almost	along	already	also
although	always	among	an	and	another
any	anybody	anyone	anything	anywhere	around
as	at	away	b	be	became
because	become	becomes	been	before	began
behind	best	better	between	both	but
by	c	can	cannot	certain	certainly
clearly	could	d	during	e	each
early	either	enough	even	evenly	ever
every	everybody	everyone	everything	everywhere	f
far	for	four	from	full	fully
further	furthered	furthering	furthers	g	general
generally	good	great	greater	greatest	h
how	however	i	if	important	in
into	is	it	its	itself	j
just	k	l	largely	later	let
lets	like	likely	m	many	may
me	might	more	most	mostly	much
n	new	necessary	no	nobody	non
noone	not	nothing	now	nowhere	o
of	off	often	on	once	only
or	other	others	our	out	over
p	per	perhaps	possible	q	quite
r	rather	really	right	s	same
shall	should	since	so	some	somebody
someone	something	somewhere	still	such	sure
t	than	that	the	their	them
then	there	therefore	these	they	thing
things	this	those	three	through	thus
to	today	too	toward	u	under
until	up	upon	us	v	video
very	w	was	way	ways	we
well	were	what	when	where	whether
which	while	who	whose	why	will
with	within	without	would	wow	x
y	yet	you	your	yours	z
all tags that are only numbers or dates					

A.3. Concept Detection

A.3.1. Average Precision per Concept

Table A.3.: A list of all Concepts and the Average Precision achieved by the concept detection pipeline.

Concept	**AP**	**Concept**	**AP**
airplane-flying	0.00963602312062	americas-got-talent	0.202761228605
anime	0.0595183176788	aquariums	0.0891902714017
arcade	0.00678667961418	asians	0.0225026170025
autumn	0.0256352734135	baby	0.0130494080408
badlands	0.0240695627657	balloons	0.0260305059727
baseball	0.0156740517021	basketball	0.0280696556358
beach	0.0194537617409	beehive	0.0766870719822
bicycle	0.00484644290852	bikini	0.00937102516569
bill-clinton	0.131920710646	birds	0.0341884446834
blacksmithing	0.092007692877	boat	0.00779682146585
boat-ship	0.0348213030872	boobs	0.0115918738257
boxing	0.194023620197	breakdancing	0.109843191303
bridge	0.0122603366161	brown-bear	0.0120870065646
bus	0.0792467424116	cake	0.102968165671
camels	0.168511561551	campus	0.00566756397367
car	0.00960971621026	car-crash	0.0134963905486
car-racing	0.0641939026657	cartoon	0.0127814714197
castle	0.00308014617674	cathedral	0.012844373003
cats	0.00698487335453	celebration	0.113885269149
cheerleading	0.0349494051903	choir	0.016040439731
christmas-tree	0.0099521634833	circus	0.0744307010655
city-skyline	0.0189369521148	cityscape	0.00610542256549
classroom	0.00672362135708	clock-tower	0.0078601257848
clouds	0.0206766614551	cockpit	0.0809616901742
commercial	0.00742543317653	concert	0.0157338343274
cooking	0.0473059215794	counterstrike-game	0.0085833623738
court	0.0526705818595	cows	0.0080486387259
crane	0.00884762663382	crash	0.0106194607392
dam	0.0210932271455	dancing	0.0173289703962
dark-skinned-people	0.0569495843294	darth-vader	0.019874658044
demonstration	0.0125326688888	desert	0.0092270970717
dog	0.0037166250243	dogs	0.00834525796795
drawing	0.137966046572	drinking	0.0069698308528
driver	0.0449732512734	drummer	0.086735750686
eiffeltower	0.0266888295276	emergency-vehicle	0.0194325363297
excavation	0.0291979813472	explosion	0.0197259983135
fence	0.00595878365814	fencing	0.295189964737
ferarri	0.0203982791604	firefighter	0.0111376976039
fireworks	0.08463040799	fish	0.0178860381735
fishing	0.0128919432713	flood	0.00627099386148
flower	0.0577270803053	food	0.0226390042705
football	0.0238575548547	forest	0.00593664283908
fountain	0.0981514460137	freeclimbing	0.0357842424012
furniture	0.00575632097579	garden	0.0102725196152
gardening	0.0157531614733	gas-station	0.00745525542456
georgewbush	0.0364067539045	geyser	0.168479410566
glacier	0.0103297699729	glasses	0.0306911704756
golf	0.053354262228	golf-course	0.747384392926

Continued on next page

Table A.3.: (Continued) A list of all Concepts and the Average Precision achieved by the concept detection pipeline.

Concept	**AP**	**Concept**	**AP**
graffiti	0.0195658124759	grand-canyon	0.0106838244804
gym	0.0143103103125	gymnastics	0.0505223124512
hand	0.0113735565671	harbor	0.223507809212
helicopter	0.0194637079755	highway	0.0672792262502
hiking	0.0105800866679	horse	0.0205948676416
horse-racing	0.0422856005036	hospital	0.00541004725881
hotel-room	0.019679612712	house	0.0113791232847
ice-skating	0.542009021181	interview	0.0129975748779
iphone	0.0177684425049	jewellery	0.125671342512
jungle	0.00801681860296	kiss	0.0081434158694
kitchen	0.0935926325584	laboratory	0.0114101361539
laundry	0.00703207089495	lava	0.0487485257616
library	0.0120323311783	lighthouse	0.00836419920614
lightning	0.0234540021991	map	0.0214472317984
marionette	0.0826571321453	market	0.00709864288251
mccain	0.164172440089	memorial	0.00566372518565
military-parade	0.0588117560145	monitor	0.0147513654651
moon	0.00917422144581	mosque	0.0119311796533
motorcycle	0.0271709219066	mountain	0.0310076722155
muppets	0.128085144056	music-video	0.0172718331189
native-american	0.037208985384	neon-sign	0.0102749212335
nighttime	0.0181141250516	obama	0.0189979907574
office	0.0240855725136	old-people	0.0136571479496
operating-room	0.0317171895336	orchestra	0.0570272294126
origami	0.46755382128	outer-space	0.0677116950319
pagoda	0.00690390632407	parachute	0.0297067552895
penguin	0.00667544565039	phone	0.0485009340956
piano	0.108418671881	pier	0.00589221899446
playground	0.00510964824151	poker	0.331616483771
polar-bear	0.0153680895562	pope	0.0634711069834
pottery	0.204431951961	press-conference	0.0176705047504
procession	0.0506388060901	pyramids	0.00480815170459
race	0.0310814872846	railroad	0.0667356168813
rainbow	0.0101389672067	rainforest	0.00958787955757
ranch	0.00566916484651	rc-car	0.0174576218778
restaurant	0.0112757137101	rice-terrace	0.0341746915622
riding	0.0268785954397	riot	0.0159688993385
river	0.00546144577815	robot	0.00825474958671
rocket-launching	0.0136327577917	rodeo	0.154448685077
rooftop	0.00474912148706	rugby	0.0818536155628
ruins	0.00941099365117	runway	0.0639273934142
safari	0.0076643928995	sailing	0.0511195431714
santa	0.00544167582092	secondlife	0.00929736178416
shipwreck	0.0524621714741	shooting	0.0387243166481
shopping-mall	0.00920758646161	simpsons	0.126208065544
singing	0.0118720417208	skateboarding	0.0111781964342
skiing	0.0393609685227	sky	0.00502751518625
snake	0.0324823989179	snooker	0.470661879116
soccer	0.0206621182619	soldiers	0.00653585082248

Continued on next page

Table A.3.: (Continued) A list of all Concepts and the Average Precision achieved by the concept detection pipeline.

Concept	**AP**	**Concept**	**AP**
stairs	0.00683438257134	steppe	0.00948485422911
street	0.00557997867973	submarine	0.00697866577619
subway	0.0712836632895	sunrise	0.0168859060336
surfing	0.0893312807198	swimming	0.0934840781057
swimming-pools	0.00583384409657	sword-fight	0.038567268664
talkshow	0.217545950365	tank	0.0384520940209
tennis	0.310807496366	tent	0.00731485163934
themepark	0.00733090949525	toilet	0.00737429949666
tony-blair	0.208274025855	tornado	0.19873383347
tractor-combine	0.0725552429714	traffic	0.00712031198206
traffic-lights	0.108325515703	tunnel	0.0175891627586
turban	0.0560585011564	two-people	0.070010906317
underwater	0.0346205057023	us-flag	0.01085319684
vending-machine	0.0280513702089	videoblog	0.0228824292074
waterfall	0.0122722935893	weather	0.446465734452
wedding	0.006931372401	wheel	0.0106350662092
windmill	0.00638727769256	windows-desktop	0.0825625729659
worldofwarcraft	0.102897192192	wrestling	0.0746893175482

A.4. Detailed Performance Comparison

A.4.1. Single Systems

Table A.4.: An overview over the performance of all single systems in comparison. The performance measure is the averaged Precision@N.

N	**History**	**Channel (all)**	**Co-Occurrence**	**Vocabulary (real)**	**Vocabulary (oracle)**	**Global**	**Nearest Neighbor**	**Channel (wc)**
1	0.457800	0.079500	0.156800	0.087100	0.604100	0.031800	0.124800	0.180800
2	0.413700	0.066300	0.155400	0.075200	0.473200	0.029300	0.113500	0.150900
3	0.378500	0.057100	0.153500	0.066900	0.393000	0.029000	0.103900	0.129900
4	0.349400	0.051100	0.145600	0.061600	0.337200	0.028000	0.097400	0.116100
5	0.326800	0.046700	0.137900	0.057100	0.297200	0.026700	0.092400	0.106400
6	0.307600	0.042600	0.131400	0.053200	0.268100	0.026300	0.088200	0.097000
7	0.290400	0.039700	0.126400	0.050400	0.245800	0.025600	0.084600	0.090400
8	0.274600	0.037200	0.120100	0.047700	0.227800	0.025300	0.080400	0.084600
9	0.260700	0.034900	0.115800	0.045900	0.213400	0.024300	0.076400	0.079500
10	0.248200	0.033000	0.111000	0.044000	0.201000	0.023300	0.073100	0.075200
11	0.236500	0.031200	0.106300	0.042200	0.189200	0.022600	0.070300	0.071100
12	0.226500	0.029800	0.102600	0.040700	0.179800	0.021900	0.067600	0.067800
13	0.217200	0.028400	0.098900	0.039300	0.171100	0.021400	0.065200	0.064600
14	0.208700	0.027100	0.096000	0.038200	0.163500	0.021000	0.063100	0.061700
15	0.201100	0.026000	0.093200	0.037000	0.156800	0.020700	0.061100	0.059200
16	0.193900	0.024900	0.090600	0.036000	0.150900	0.020400	0.059300	0.056700
17	0.187300	0.024000	0.088100	0.035200	0.145300	0.020100	0.057600	0.054500
18	0.181200	0.023100	0.085900	0.034100	0.140200	0.019800	0.056000	0.052700
19	0.175600	0.022500	0.083600	0.033200	0.135800	0.019500	0.054500	0.051100
20	0.170300	0.021800	0.081600	0.032400	0.131300	0.019300	0.053200	0.049600
21	0.165400	0.021200	0.079800	0.031700	0.127500	0.019000	0.051900	0.048200
22	0.160800	0.020600	0.078000	0.031000	0.123700	0.018600	0.050800	0.047000
23	0.156400	0.020100	0.076400	0.030500	0.120100	0.018400	0.049800	0.045600
24	0.152300	0.019500	0.074800	0.029800	0.116900	0.018100	0.048800	0.044400
25	0.148500	0.019000	0.073300	0.029200	0.113900	0.018000	0.047800	0.043300

Table A.5.: An overview over the performance of all single systems in comparison. The performance measure is the averaged Recall@N.

N	**History**	**Channel (all)**	**Co-Occurrence**	**Vocabulary (real)**	**Vocabulary (oracle)**	**Global**	**Nearest Neighbor**	**Channel (wc)**
1	0.053500	0.008100	0.012700	0.010100	0.074200	0.002800	0.012000	0.018500
2	0.092100	0.013100	0.025200	0.016200	0.110000	0.004700	0.021300	0.029900
3	0.120600	0.016800	0.037400	0.020500	0.131600	0.007000	0.028400	0.038300
4	0.143600	0.019500	0.046900	0.024100	0.146200	0.008800	0.034900	0.044400
5	0.162700	0.022100	0.055100	0.027100	0.157400	0.010300	0.040600	0.050200
6	0.179100	0.023800	0.062600	0.029800	0.167200	0.012500	0.045700	0.054200
7	0.192900	0.025600	0.069500	0.032200	0.176100	0.014200	0.050300	0.058200
8	0.205100	0.027100	0.075100	0.034400	0.184000	0.016100	0.054100	0.061700
9	0.216000	0.028500	0.080900	0.036600	0.191200	0.017200	0.057300	0.064700
10	0.225600	0.029800	0.085800	0.038500	0.197700	0.018200	0.060400	0.067800
11	0.233700	0.030800	0.089700	0.040400	0.202900	0.019400	0.063400	0.070100
12	0.241400	0.031900	0.093900	0.041900	0.208400	0.020500	0.066100	0.072600
13	0.248500	0.032700	0.097600	0.043400	0.213000	0.021400	0.068500	0.074500
14	0.255100	0.033500	0.101700	0.045200	0.217700	0.022900	0.070800	0.076300
15	0.261000	0.034300	0.105100	0.046600	0.222000	0.023700	0.072800	0.078000
16	0.266100	0.034900	0.108300	0.048100	0.226400	0.024900	0.074800	0.079300
17	0.271100	0.035400	0.111500	0.049500	0.230100	0.026300	0.076800	0.080600
18	0.275800	0.036100	0.114600	0.050600	0.233600	0.027000	0.078800	0.082200
19	0.280200	0.036900	0.117500	0.051700	0.237500	0.028200	0.080500	0.083800
20	0.284500	0.037500	0.120300	0.052800	0.240600	0.029800	0.082300	0.085400
21	0.288200	0.038200	0.123000	0.053900	0.244300	0.030700	0.084000	0.087000
22	0.292000	0.038800	0.125400	0.055000	0.247300	0.031400	0.085800	0.088400
23	0.295400	0.039400	0.128200	0.056200	0.250100	0.032300	0.087700	0.089600
24	0.298800	0.040100	0.130400	0.057200	0.252800	0.033000	0.089500	0.091200
25	0.302000	0.040600	0.132900	0.058200	0.255500	0.034100	0.091100	0.092500

A.4.2. Fused Systems

Table A.6.: An overview over the performance of all fused systems in comparison. The performance measure is the averaged Precision@N.

N	**History**	**Rule (11)**	**PersonalizedTagTransfer (0.7)**	**WeightedSum (oracle)**	**WeightedSum (learned)**
1	0.457800	0.462400	0.474900	0.571500	0.459900
2	0.413700	0.417700	0.430800	0.516600	0.406400
3	0.378500	0.382100	0.393400	0.471100	0.362700
4	0.349400	0.352700	0.364400	0.435000	0.336100
5	0.326800	0.329800	0.340000	0.406000	0.316300
6	0.307600	0.310300	0.320000	0.381100	0.299100
7	0.290400	0.293100	0.303000	0.359400	0.283600
8	0.274600	0.277100	0.286900	0.339900	0.269200
9	0.260700	0.263100	0.272500	0.322400	0.256300
10	0.248200	0.250500	0.259600	0.306800	0.244800
11	0.236500	0.238700	0.247900	0.292900	0.234500
12	0.226500	0.228500	0.237300	0.280000	0.225000
13	0.217200	0.219200	0.227600	0.268200	0.216500
14	0.208700	0.210700	0.218800	0.257700	0.208700
15	0.201100	0.203000	0.210900	0.248100	0.201600
16	0.193900	0.195700	0.203400	0.239200	0.194900
17	0.187300	0.189100	0.196500	0.230800	0.188600
18	0.181200	0.183000	0.190100	0.223100	0.183000
19	0.175600	0.177200	0.184200	0.215700	0.177600
20	0.170300	0.172000	0.178800	0.209000	0.172700
21	0.165400	0.167000	0.173700	0.202700	0.168100
22	0.160800	0.162300	0.168900	0.196600	0.163700
23	0.156400	0.157900	0.164300	0.190900	0.159500
24	0.152300	0.153800	0.160100	0.185500	0.155600
25	0.148500	0.150000	0.156200	0.180300	0.151900

Table A.7.: An overview over the performance of all fused systems in comparison. The performance measure is the averaged Recall@N.

N	**History**	**Rule (11)**	**PersonalizedTagTransfer (0.7)**	**WeightedSum (oracle)**	**WeightedSum (learned)**
1	0.053500	0.053900	0.055600	0.067700	0.053900
2	0.092100	0.092900	0.095700	0.116400	0.090500
3	0.120600	0.121700	0.125400	0.152300	0.115200
4	0.143600	0.144900	0.149700	0.181600	0.137600
5	0.162700	0.164200	0.169600	0.205800	0.158200
6	0.179100	0.180900	0.187100	0.225900	0.175200
7	0.192900	0.194800	0.202400	0.243300	0.189900
8	0.205100	0.207200	0.215400	0.258400	0.202700
9	0.216000	0.218200	0.227000	0.271600	0.214100
10	0.225600	0.227900	0.237300	0.283300	0.224500
11	0.233700	0.236000	0.246500	0.293900	0.234200
12	0.241400	0.243800	0.254700	0.302900	0.242600
13	0.248500	0.251100	0.262200	0.311100	0.250700
14	0.255100	0.257700	0.268900	0.318700	0.258200
15	0.261000	0.263700	0.275300	0.325800	0.265100
16	0.266100	0.268900	0.281000	0.332100	0.271500
17	0.271100	0.273900	0.286200	0.337900	0.277000
18	0.275800	0.278700	0.291400	0.343700	0.282700
19	0.280200	0.283300	0.296200	0.348400	0.287900
20	0.284500	0.287600	0.301000	0.353300	0.293200
21	0.288200	0.291400	0.305500	0.357700	0.297800
22	0.292000	0.295200	0.309500	0.361700	0.302300
23	0.295400	0.298700	0.313400	0.365500	0.306600
24	0.298800	0.302200	0.317200	0.368800	0.310300
25	0.302000	0.305500	0.320800	0.371900	0.314000

B. List of Figures

C. List of Tables