

UNIVERSITÀ DEGLI STUDI DI PADOVA

DEPARTMENT OF INFORMATION ENGINEERING

Master Thesis in Computer Engineering

**S-AWARE: MEASURE-BASED
SUPERVISED MERGING ALGORITHMS
FOR CROWD ASSESSORS IN
INFORMATION RETRIEVAL**

Supervisor

PROF. NICOLA FERRO

Master Candidate

LUCA PIAZZON

*A.Y. 2018-2019
9 December 2019*

*This thesis work is
dedicated to the memory
of my grandfather Vittorio,
who always believed in me.*

ABSTRACT

In the field of information retrieval, one of the most important tasks is evaluating the performance of IR systems. In order to do this, a trusted ground truth is needed. In the last years, crowdsourcing has began to be used as a viable and cheap alternative to experts' ground truth. In this thesis we develop a new supervised approach to exploit crowd assessors relevance judgements for information retrieval evaluation. Our work continues the research started in the last years on AWARE probabilistic framework. While the common state of the art methods aim to create a single ground truth from the assessors' judgements, in our approach we compute evaluation measures based on the ground truth generated from each crowd assessor. These measures are then merged weighting each assessor on the basis of his expertise level. In our approach, assessor expertise estimation is obtained in a supervised way analysing the closeness between the measures computed on assessors' judgements and the measures computed on experts' judgements, on a training set. Such closeness measure has been computed following several different methodologies. We tested our approaches against some classic approaches and a set of unsupervised approaches from the u-AWARE framework, considering different combinations of evaluation measure, set of IR systems and number of merged assessors. Test results highlight the greater effectiveness of our supervised approaches with respect to the majority of the approaches. Additionally, the impact of the training-set size on performance has been studied, stating that even with a small training-set is possible to achieve good results.

TABLE OF CONTENTS

	Page
List of Figures	vii
List of Algorithms	xi
List of Tables	xiii
1 Introduction	1
2 Background and Related Work	5
2.1 Information retrieval	5
2.2 IR evaluation	7
2.2.1 Cranfield paradigm and evaluation campaigns	7
2.2.2 Test collection definition	8
2.2.3 Evaluation measures	10
2.2.3.1 Precision and Recall	10
2.2.3.2 Average Precision	10
2.2.3.3 Normalized Discounted Cumulative Gain	11
2.3 Collective intelligence, Human computation and Crowdsourcing	12
2.3.1 Crowdsourcing platforms	14
2.3.2 Games with a purpose	14
2.4 Crowdsourcing in IR	16
2.4.1 Crowdsourcing for relevance evaluation	16
2.4.2 Noisy judgements	18
2.5 Crowdsourcing techniques	19
2.5.1 Majority vote	20
2.5.2 Expectation maximization	22
2.5.3 GeAnn	23

TABLE OF CONTENTS

2.5.4	TurkRank	24
2.5.5	Skierarchy	25
3	AWARE framework	27
3.1	Dissimilarity definitions	30
3.1.1	Measure dissimilarity	30
3.1.2	Distribution dissimilarity	31
3.1.3	Ranking dissimilarity	31
3.2	u-AWARE accuracy computation	32
3.2.1	GAP	33
3.2.2	Weight	35
3.3	s-AWARE accuracy computation	39
3.3.1	GAPs and normalization	40
4	Experimental Setup	43
4.1	Experimental parameters	44
4.1.1	Dataset	44
4.1.1.1	Crowd assessors collection	44
4.1.1.2	Retrieval Systems	46
4.1.2	Evaluation measures	46
4.1.3	Analysis measures	47
4.1.4	Parameters	48
4.1.4.1	Topicsets	48
4.1.4.2	Kuples	50
4.1.4.3	Other parameters	50
4.2	Experimental workflow	51
4.2.1	Data import	53
4.2.2	Base measures	54
4.2.3	Classic Approaches	57
4.2.4	u-AWARE and s-AWARE approaches	60
4.2.5	ANOVA analysis	65
5	Experimental Results	71
5.1	Base measures analysis	71
5.1.1	AP Correlation	71

5.1.2	RMSE	73
5.2	Results with equal Train-Test size	74
5.2.1	AP Correlation	74
5.2.2	RMSE	79
5.3	Results with different topicset sizes	84
6	Conclusions and Future work	87
6.1	Future Work	89
A	Plots and ANOVA tables	91
A.1	Results with topicset of 3 topics	91
A.1.1	AP Correlation	91
A.1.2	RMSE	94
A.2	Results with topicset of 7 topics	96
A.2.1	AP Correlation	96
A.2.2	RMSE	99
	Bibliography	103

LIST OF FIGURES

FIGURE	Page
2.1 Information retrieval process schema	6
2.2 Pooling technique	9
2.3 Amazon Mturk	14
2.4 Peakaboom Game interface	15
2.5 Relevance evaluation task on MTurk	17
2.6 Reasons for Crowd-expert disagreement	18
2.7 screen view of the geAnn game	24
2.8 Example of network model, showing crowd assessors agreements between crowd assessors and with gold standard	25
2.9 Hierarchy of assessors used in Skierarchy	26
3.1 Classic approaches methodology vs AWARE methodology	28
3.2 u-AWARE accuracy is computed with dissimilarity measures be- tween crowd assessors and different types of random assessors . . .	32
3.3 AWARE GAP-Weight combinations	33
3.4 $ T \times S $ matrix for the assessor measures	34
3.5 s-AWARE accuracy is computed with dissimilarity measures be- tween crowd assessors and the gold standard on a training topicset	39
4.1 Experimental workflow	51
4.2 Data import: pools and runs are organized into data structures . . .	53
4.3 Base measure computation: for each assessor k , measures on runs are computed taking the assessor's judgements as gold standard . .	54
4.4 Base measure analysis: measures are averaged by topic and anal- ysed with AP Correlation and RMSE	55
4.5 APC and RMSE average over the topicsets	55

4.6	Pool merging: a single merged pool is computed using the relevance judgements coming from all the assessors	57
4.7	Pool measure computation: for each kuple, measures are computed taking the merged pool of the kuple as ground truth	58
4.8	Pool measure analysis: measures are averaged by topic and analysed with AP Correlation and RMSE	58
4.9	APC and RMSE average over topicsets	59
4.10	AWARE accuracy scores computation: GAPs are computed with respect to each random replicate. The mean GAPs are then combined with Weight computation	60
4.11	s-AWARE accuracy scores computation: normalized GAPs are directly used as accuracy scores	61
4.12	AWARE measure computation: merged measures are computed weighting assessors' by the accuracy scores	61
4.13	AWARE measures analysis: measures are averaged by topic and analysed with AP Correlation and RMSE	62
4.14	APC and RMSE average over topicsets	62
4.15	Different values of Y , relative to different combinations of topics and factor A values	66
4.16	ANOVA analysis	69
5.1	APC for base measures	72
5.2	APC Assessor*Measure interaction for base measures	72
5.3	APC Assessor*Runset interaction for base measures	72
5.4	RMSE for base measures	73
5.5	RMSE Assessor*Measure interaction for base measures	74
5.6	RMSE Assessor*Runset interaction for base measures	74
5.7	APC Approach main effect (5 test topics)	75
5.8	APC Kuple main effect (5 test topics)	76
5.9	APC Measure main effect (5 test topics)	77
5.10	APC Runset main effect (5 test topics)	77
5.11	APC Approach*Kuple interaction effect (5 test topics)	78
5.12	APC Approach*Measure interaction effect (5 test topics)	79
5.13	APC Approach*Runset interaction effect (5 test topics)	79
5.14	RMSE Approach main effect (5 test topics)	80

5.15 RMSE Kupple main effect (5 test topics)	81
5.16 RMSE Measure main effect (5 test topics)	81
5.17 RMSE Runset main effect (5 test topics)	81
5.18 RMSE Approach*Kupple interaction effect (5 test topics)	83
5.19 RMSE Approach*Measure interaction effect (5 test topics)	84
5.20 RMSE Approach*Runset interaction effect (5 test topics)	84
5.21 AP Correlation of approaches for different topicset sizes	84
5.22 RMSE of approaches for different topicset sizes	85
A.1 APC Approach main effect (3 test topics)	92
A.2 APC Measure main effect (3 test topics)	92
A.3 APC Kupple main effect (3 test topics)	92
A.4 APC Runset main effect (3 test topics)	92
A.5 APC Approach*Kupple interaction effect (3 test topics)	93
A.6 APC Approach*Measure interaction effect (3 test topics)	93
A.7 APC Approach*Runset interaction effect (3 test topics)	93
A.8 RMSE Approach main effect (3 test topics)	94
A.9 RMSE Measure main effect (3 test topics)	94
A.10 RMSE Kupple main effect (3 test topics)	95
A.11 RMSE Runset main effect (3 test topics)	95
A.12 RMSE Approach*Kupple interaction effect (3 test topics)	95
A.13 RMSE Approach*Measure interaction effect (3 test topics)	96
A.14 RMSE Approach*Runset interaction effect (3 test topics)	96
A.15 APC Approach main effect (7 test topics)	97
A.16 APC Measure main effect (7 test topics)	97
A.17 APC Kupple main effect (7 test topics)	97
A.18 APC Runset main effect (7 test topics)	97
A.19 APC Approach*Kupple interaction effect (7 test topics)	98
A.20 APC Approach*Measure interaction effect (7 test topics)	98
A.21 APC Approach*Runset interaction effect (7 test topics)	98
A.22 RMSE Approach main effect (7 test topics)	99
A.23 RMSE Measure main effect (7 test topics)	99
A.24 RMSE Kupple main effect (7 test topics)	100
A.25 RMSE Runset main effect (7 test topics)	100
A.26 RMSE Approach*Kupple interaction effect (7 test topics)	100

LIST OF FIGURES

A.27 RMSE Approach*Measure interaction effect (7 test topics)	101
A.28 RMSE Approach*Runset interaction effect (7 test topics)	101

LIST OF ALGORITHMS

ALGORITHM	Page
1 How to compute sgl accuracy with u-AWARE	37
2 How to compute tpc accuracy with u-AWARE	38
3 How to compute Assessor scores with s-AWARE	42
4 Import collection pseudocode	54
5 Compute base measures pseudocode	56
6 Classic approaches pipeline pseudocode	59
7 u-AWARE approaches pipeline pseudocode	63
8 s-AWARE approaches pipeline pseudocode	64

LIST OF TABLES

TABLE	Page
4.1 Description for topics used in TREC 21 Crowdsourcing Track	45
4.2 Gold standard relevance judgements. From left to right: topic id, the total number of assessed documents by TREC participants, the number of NIST relevance judgements in the pool and its fraction, the number of NIST/assessors disagreements and its fraction, the number of NIST relevant documents finally labelled as non relevant, the number of NIST non relevant documents finally labelled as relevant, the total relevant documents per topic after the process .	46
4.3 List of the crowd Pools used un the experiments	47
4.4 topicsets used in experiments with 7 training topics and 3 test topics	49
4.5 topicsets used in experiments with 5 training topics and 5 test topics	49
4.6 topicsets used in experiments with 3 training topics and 7 test topics	50
5.1 ANOVA table for AP Correlation (5 test topics)	75
5.2 ANOVA table for RMSE (5 test topics)	80
5.3 assessors average agreement with gold standard considering the top 20 documents for each run	86
A.1 ANOVA table for AP Correlation (3 test topics)	91
A.2 ANOVA table for RMSE (3 test topics)	94
A.3 ANOVA table for AP Correlation (7 test topics)	96
A.4 ANOVA table for RMSE (7 test topics)	99

INTRODUCTION

In Information Retrieval, the problem of creating test collections that correctly represent the reference scenario is crucial, in order to evaluate information retrieval systems performance and improve them.

The concept of test collection has been introduced in '60s by Ceryl Cleverdon [1]: given a set of documents and a set of topics, a relevance judgement is assigned to each document for each topic. This evaluation process is historically been performed by an expert team following the Cranfield paradigm, which is, however, very economically demanding and time-consuming.

A modern approach to this task, in the field of crowdsourcing, is based on the idea to collect and combine judgements from many crowd assessors, less qualified than the experts but cheaper. The objective is to achieve a ground truth as similar as possible to the expert team ground truth.

To collect this data, several crowdsourcing platforms, as Amazon Mechanical Turk, are now available: organizations which ask for the crowd assessment can post tasks on the platform, where users can find and perform them in exchange for a reward.

Exploiting this new opportunity, in the last years many different approaches have been developed in order to merge multiple crowd judgements [2–8].

Classic state of the art methods, given a set of crowd assessors' judgements, aim to create a merged ground truth from the judgements given by all the

assessors. The most common approaches are Majority Vote, that creates a ground truth assigning to each document the most popular judgement among crowd assessors, and Expectation Maximization, that iteratively estimates until convergence the document probability of relevance, and then assigns to each document the most probable judgement.

In this thesis, a different methodology has been followed, according to that used in the paper “AWARE: Exploiting Evaluation Measures to Combine Multiple Assessors”[9]: the basic idea is not to combine crowd assessors’ relevance judgements in a single ground truth, but to evaluate IR systems on the judgements given by every single assessor, and then combine the obtained measures weighting each assessor on the basis of an estimation of their expertise level. In the original AWARE paper [9], expertise level is estimated in an unsupervised way using some dissimilarity measures, called GAPs, between evaluation measures computed using assessors’ judgements as ground truth and measures based on three dummy random assessors: a large GAP means a not-random behaviour and then a high level of assessor’s expertise.

In the approaches subject of this thesis, called s-AWARE (supervised-AWARE), assessors’ expertise is obtained in a supervised way looking at their performance on a training set of topics. Some dissimilarity scores are calculated between the evaluation measures computed on assessors’ judgements and the same evaluation measures computed on the experts’ ground truth: a smaller dissimilarity value means a higher expertise level.

This thesis work is about the development of such methods and their comparison against the approaches presented in [9], here called u-AWARE, Majority Vote and Expectation Maximization.

Experiments are computed considering two different sets of runs, two different evaluation measures and many different cardinalities for the set of assessors to be merged (from 2 to 30): all the combinations are tested in order to determine how the different approaches behave with respect to each parameter.

To compare approaches performance, RMSE and AP-Correlation are computed between the merged measures obtained by each approach and the measures obtained evaluating IR systems on experts’ ground truth.

The thesis chapters are structured as follows:

- Chapter 2 - Background: in the first two sections, we introduce the reader to information retrieval and its main definitions and we describe the evaluation measures that will be used in the experiments. In the rest of the chapter we give an overview on crowdsourcing and its application in information retrieval evaluation, reporting some of the state of the art approaches
- Chapter 3 - AWARE Framework: we describe AWARE motivations and methodologies and we explain our proposed approaches
- Chapter 4 - Experimental Setup: we describe experimental parameters and the workflow of the experiments
- Chapter 5 - Experimental Results: we analyse the results of our experiments, highlighting how s-AWARE approaches behave with respect to all the other approaches
- Chapter 6 - Conclusions and Future Work: we summarize the results and we outline possible future improvements of the s-AWARE framework

BACKGROUND AND RELATED WORK

2.1 Information retrieval

Information retrieval is a part of information science which studies methods to obtain relevant resources from a data collection, in order to satisfy an information need from an user.

A more formal definition is given by Gerald Salton in 1968 [10]: "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."

The term "information" is very general and includes both text documents (web pages, papers, books, articles,...) and multimedia content (music, images, videos).

The main differences between Information Retrieval Systems (IRS) and database management systems (DBMS) are about:

- Data structure: databases store data in a structured way, IR Systems work with unstructured data, often using natural language
- Queries: database queries are unambiguous, IR queries are not
- Result quality: Data from databases is always correct in a formal sense, because is unequivocally represented by the query; data retrieved by IRSs might be or not to be relevant for the user

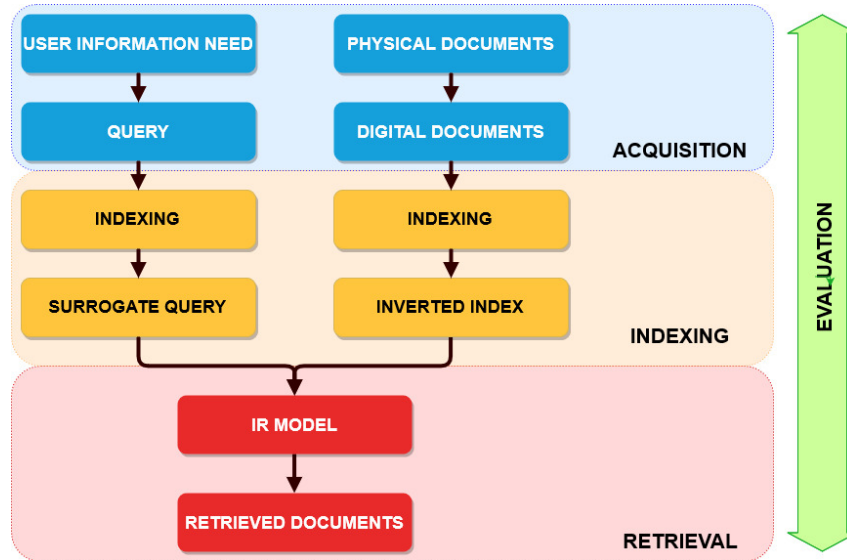


Figure 2.1: Information retrieval process schema

The information retrieval process [10] consists of four main steps, shown in Figure 2.1:

- **Acquisition:** Physical documents are digitalized and organized in collections, user's information need is represented with a query
- **Indexing:** Documents are processed to create an inverted index, a data structure in which all the terms are connected to the documents containing them. In order to create such an index, each document is initially analysed to detect tokens (i.e. terms in text documents), then a list of common words (called stopwords) is removed to keep only terms with high information content. The resulting set of terms can then be processed with stemming techniques, which replace each term with his root (e.g. information, informed, informative,... → inform). Some terms which are frequently used together are then composed (e.g. "inform retrieve" is more specific than the two terms taken separately). At the end of the process, a weight is assigned to each term-document pair, based on the frequency of the term in the document.
- **Retrieval:** a retrieval model is used to find the most relevant documents to the user query. To do this, the IRS matches query terms and inverted index terms, searching for the best match.

- **Evaluation:** to rate the effectiveness of the system, some evaluation metrics are computed. In the next section we'll introduce the concept of relevance and describe the main evaluation metrics that will be used in the thesis experiments.

2.2 IR evaluation

2.2.1 Cranfield paradigm and evaluation campaigns

The purpose of IR evaluation is to compute effectiveness measures of IRs comparing them and identifying strengths and weaknesses to be used to improve systems performance. All the components of the IR pipeline contribute to produce the ranked list of documents which will be evaluated, so the evaluation process is related to all the IR process steps in Figure 2.1.

The standard for evaluation in IR is the Cranfield paradigm, developed in '60 by Ceryl Cleverdon, which introduced the concept of experimental collection. The first experiment [11], called Cranfield 1, was run to test four different manual indexing methods over a collection of 18000 articles and papers. Each document has been indexed by three experts, taking two years to complete the process. Each index has then been used to retrieve documents based on some simple queries written by the articles' authors.

At the end of this retrieval phase, the results highlight that 35% of the queries didn't retrieve the correct document [1]: analysing the possible causes, Cleverdon understood that some errors were done by the experts while choosing documents descriptors in indexing phase.

A second experiment, named Cranfield 2, has been run in a more structured way to further investigate the indexing methods effectiveness. The main difference with the Cranfield 1, is that documents and topics (i.e queries) has now been selected, creating a collection of 1400 documents and 221 queries (with its own sets of relevance judgements on documents) faithfully representing the domain of interest. Tests were run on 33 different indexing methods, and results show that indexing methods based single terms perform better of methods based on term combinations. This new method of running tests allows to reuse collections for different experiments or to reproduce past experiments. In the last decades, the Cleverdon methodology has been followed in several

evaluation Campaigns, whose purpose is creating test collections in a collaborative way between different research groups.

Nowadays, the main evaluation campaigns are TREC ¹(Text REtrieval Conference) in the USA, CLEF ² (Conference and Labs of Evaluation Forum) in Europe, FIRE ³(Forum for Information Retrieval Evaluation) in India and NTCIR ⁴(NII Testbeds and Community for Information access Research) in Asia.

2.2.2 Test collection definition

According to what has just been said, we define [12] a test collection as a triple $C = \{D, T, GT\}$ where:

- $D = \{d_1, d_2, \dots, d_n\}$ is a set of documents
- $T = \{t_1, t_2, \dots, t_m\}$ is a set of topics
- GT is the ground truth.

To understand what the Ground Truth is, we first introduce the concept of relevance. Relevance is a property that represent the capability of a document to satisfy an user information need: relevance can be defined as a binary property or a multi-graded property.

Let REL be a finite set of relevance degrees and let \leq be a total order relation on REL so that (REL, \leq) is a totally ordered set.

We call non-relevant the relevance degree $nr \in REL$ such that $nr = \min(REL)$.

The Ground truth is then defined as the function that assigns a relevance judgement to every topic-document pair, formally:

Let D bet a finite set of documents and T a finite set of topics. The ground truth is the function

$$GT : T \times D \rightarrow REL$$

$$(t, d) \rightarrow rel$$

¹<https://trec.nist.gov/>

²<http://www.clef-initiative.eu/>

³<http://fire.irsir.res.in>

⁴<http://research.nii.ac.jp/ntcir/index-en.html>

Assessing each document for every topic would be a too long process: the standard approach used in TREC to reduce the amount of necessary relevance judgements is to use pooling.

To explain pooling is first necessary to introduce the concept of run as a function that assigns to each topic a ranked list of documents. Given a natural number $N \in \mathbb{N}^+$ called run length, a run is defined as the function:

$$R : T \rightarrow D^N$$

$$t \rightarrow r_t = (d_1, d_2, \dots, d_n)$$

such as $\forall t \in T, \forall j, k \in [1, N] \parallel j \neq k \Rightarrow r_t[j] \neq r_t[k]$ where $r_t[j]$ is the j -th element in the vector r_t .

In Figure 2.2 is represented the pooling technique: given a set of runs for the same topic, pooling consist of assessing only the top- k documents for each run. All the other documents are considered not relevant for the given topic.

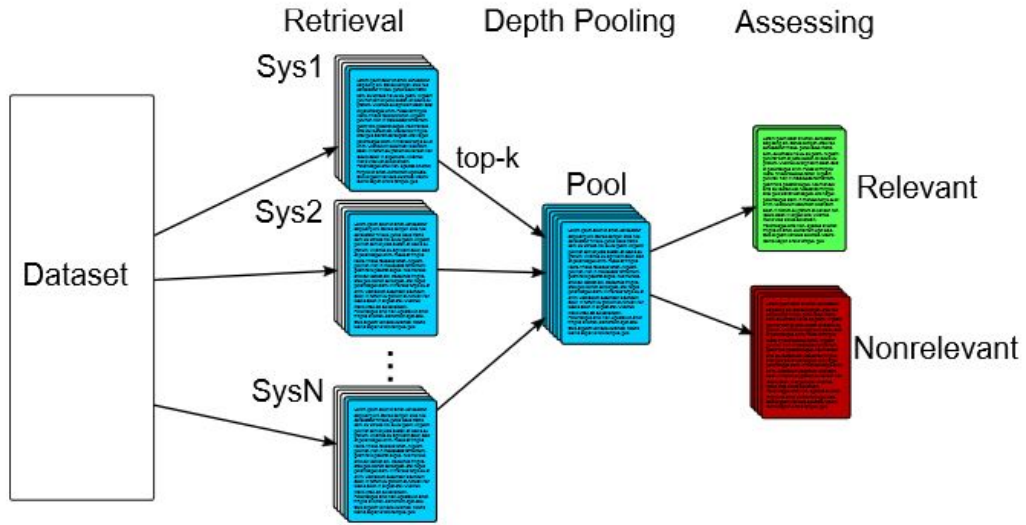


Figure 2.2: Pooling technique

2.2.3 Evaluation measures

In this section, we will describe the most common evaluation measures as reported in [12, 13].

2.2.3.1 Precision and Recall

The two main goals of IRS are finding all possible relevant documents and retrieving only relevant documents. Two simple measures are defined to evaluate a run based on the capability of achieving such properties.

Let D^* be the set of relevant documents for a given topic, and let D be the set of all the documents retrieved by an IRS for the same topic. We define Precision as the fraction of relevant documents over all the retrieved documents:

$$Prec = \frac{|D^* \cap D|}{|D|}$$

We define Recall as the fraction of relevant documents retrieved by the IRS:

$$Rec = \frac{|D^* \cap D|}{|D^*|}$$

where $|D^*|$ is also called Recall Base (RB) for the topic.

Recall and Precision can be adapted to ranked lists of documents, restricting to the first k retrieved documents: we talk therefore respectively of $Rec@k$ and $Prec@k$.

2.2.3.2 Average Precision

Precision and recall represent the two sides of the coin: trying to improve precision, it is likely to worsen recall and vice versa. Average precision is one of the most used measures as it tries to summarize both properties in a top-heavy measure, meaning that runs with relevant documents on the top of the list are judged better than the others.

Given a topic $t \in T$, a recall base RB_t , $REL = \{nr, r\}$, a run r_t of size $N \in \mathbb{N}^+$ with relevance judgements \hat{r}_t such that relevance weights are defined as:

$$\forall i \in [1, N], \tilde{r}_t = \begin{cases} 0, & \text{if } \hat{r}_t[i] = nr \\ 1, & \text{if } \hat{r}_t[i] = r \end{cases}$$

we can define Average Precision (AP) as :

$$AP = \frac{1}{RB_t} \sum_{k=1}^N \tilde{r}_t[k] \frac{\sum_{h=1}^k \tilde{r}_t[h]}{k}$$

We can then define Mean Average Precision (MAP) as the mean of AP over the topics:

$$MAP = \frac{\sum_{t \in T} AP(r_t)}{|T|}$$

2.2.3.3 Normalized Discounted Cumulative Gain

Another family of measures is based on Cumulative Gain (CG): Let r_t be a run of size $N \in \mathbb{N}^+$, where $t \in T$ is a topic, and consider $j \in \mathbb{N}^+ \mid 1 \leq j \leq N$. Let also \tilde{r}_t be the relevance weights for the documents in the run, then the Cumulative Gain at rank position j is defined as:

$$CG[j] = \sum_{k=1}^j \tilde{r}_t[k]$$

A top heavy version of CG is called Discounted Cumulative Gain (DCG) and uses a discounting function to progressively reduce document weight as the ranking decrease. Given a run r_t of size $N \in \mathbb{N}^+$ and $b \in \mathbb{N}^+$, we first define discounted gain as:

$$dg_{r_t}^b[k] = \begin{cases} \tilde{r}_t[k] & \text{if } k < b \\ \frac{\tilde{r}_t[k]}{\log_b k} & \text{otherwise} \end{cases} \quad \forall k \in [1, N]$$

Discounted cumulative gain can then be defined as:

$$DCG[j] = \sum_{k=1}^j dg_{r_t}^b[k]$$

DCG, as CG, is not a limited measure and the maximum value may be different for every topic. To obtain a measure in range $[0, 1]$, we introduce normalized Discounted Cumulative Gain, a measures that compares DCG with the DCG obtained on the ideal run.

The ideal run $i(t)$ is the run where all relevant documents are retrieved with the best possible ranking, and represents the perfect retrieval scenario for a given topic. Normalized Discounted Cumulative Gain is defined as:

$$nDCG[j] = \frac{\sum_{k=1}^j dg_{r_t}^b[k]}{\sum_{k=1}^j dg_{i_t}^b[k]}$$

2.3 Collective intelligence, Human computation and Crowdsourcing

"The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer."[14, 15]: this was the Alan Turing forecast about computers in 1950. After a few decades, we can say that there are still tasks that can be performed by a human and are impossible for digital computers, or in which humans largely outperform computers.

Therefore, with the development of Information Technology, the more humans depend on computers the more computers need human computation, defined by von Ahn in 2005 as "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve"[15, 16]. Human computation includes several different tasks in fields as artificial intelligence, cryptography, genetic algorithms and in general human-computer interaction [15].

In this thesis, we focus on crowdsourcing, a concept related both to human computation and collective intelligence, the intelligence generated by the interaction of multiple collaborating and cooperating agents, on the basic idea that "no one knows everything, each one knows something".

The term crowdsourcing was first used by Jeff Howe in 2006 [17], as the union of the two words "crowd" and "outsourcing": "Crowdsourcing is the act of taking a job traditionally performed by a designated agent and outsourcing it to an undefined, generally large group of people in the form of an open call".

The objective of crowdsourcing is leveraging the wisdom of the crowds to compute tasks in a more convenient way, trying to achieve equal or better performance with respect to the designated agent[18].

Crowdsourcing should not be confused with open-source production: in crowdsourcing, people are invited to respond to activities promoted by an organization and they are motivated to respond for a variety of reasons, in open-source projects, instead, there is no need for human computation by an organization and the work is performed and utilized only by a community of users [19, 20]. An example of crowdsourcing is Threadless, an online clothing company based on a community of artists that create, submit and evaluate designs for the products to be realized and put on the market. Designers are economically

rewarded for every accepted project, and the company benefits of distributed and fast work, saving on hiring professional designers. The classical example of what crowdsourcing is not is Wikipedia, a free online encyclopedia written collaboratively by volunteers, where Wikipedia organization does not indicate what articles need to be written.

Crowdsourcing applications in information technology are often related to artificial intelligence: in this field, crowdsourcing has become a valid method to obtain large amounts of labelled data to use for the training of algorithms [21–23].

The two main questions about crowdsourcing are how to motivate people to perform a task and how to control the crowd workers performance.

Main motivation factors can be:

- Monetary reward (e.g. in Crowdsourcing platforms: 2.3.1)
- Enjoyment (e.g in Games with a purpose: 2.3.2)
- Public reputation
- Integration in other processes (e.g. ReCAPTCHA: [24])

Some methods to control work quality can be:

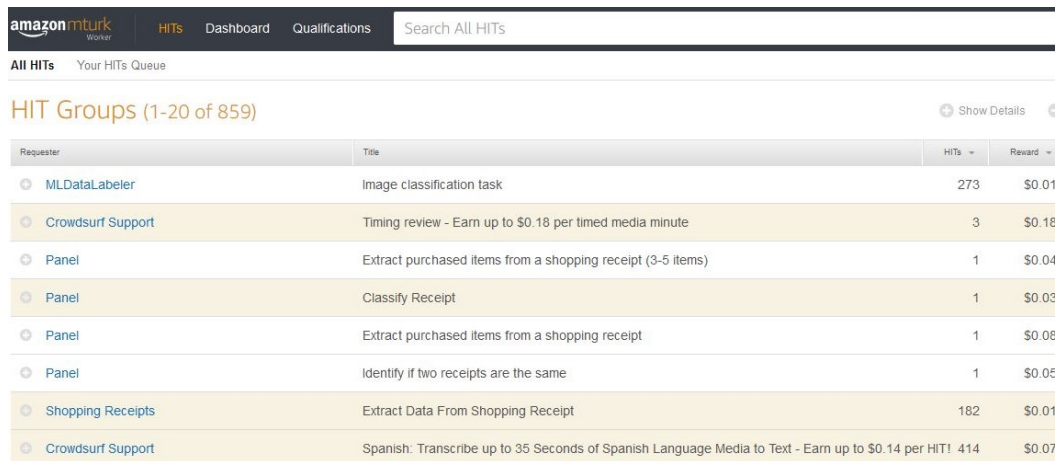
- Redundancy: the task is assigned to more than one user, to detect poor answers.
- Ground truth seeding: the first few assigned tasks are used as a quality test of the worker, comparing the worker's answers with a trusted source.
- Multilevel review: the work done by a set of workers is then reviewed multiple times by other users.
- Expert review: some submissions are manually reviewed to check workers' reliability.
- Automatic check: some tasks are difficult to compute but can be automatically checked with a minimal computational effort.
- Justification: in the case of subjective tasks, workers are required to give a justification for their answers, to discourage spam answers.

- Expertise check: some simple questions about the topic are asked to the worker, to guarantee a minimum degree of knowledge

2.3.1 Crowdsourcing platforms

Most of the possible applications of crowdsourcing require the repetition of simple operations: the smallest unit of work to be performed is called Human Intelligence Task (HIT). Requesters, which can be individuals or organizations, post HITs on crowdsourcing platforms, where crowd workers can find them and perform tasks in exchange for a monetary reward.

The most popular crowdsourcing platforms are called Figure Eight ⁵ and Amazon Mechanical Turk ⁶, which has more than 100 thousand active users from 190 different countries [25].



Requester	Title	HITs	Reward
MLDataLabeler	Image classification task	273	\$0.01
Crowdsurf Support	Timing review - Earn up to \$0.18 per timed media minute	3	\$0.18
Panel	Extract purchased items from a shopping receipt (3-5 items)	1	\$0.04
Panel	Classify Receipt	1	\$0.03
Panel	Extract purchased items from a shopping receipt	1	\$0.08
Panel	Identify if two receipts are the same	1	\$0.05
Shopping Receipts	Extract Data From Shopping Receipt	182	\$0.01
Crowdsurf Support	Spanish: Transcribe up to 35 Seconds of Spanish Language Media to Text - Earn up to \$0.14 per HIT!	414	\$0.07

Figure 2.3: Amazon Mturk

2.3.2 Games with a purpose

Another way to get useful information from the crowd is to collect data in Games With a Purpose (GWAP). These online games are designed to drive the user to complete intelligent tasks without him noticing. Luis von Ahn first proposed the idea of these games [26]: his approach was based on the competition between two users, so that every user is driven to give good answers.

Some of the main tasks in which GWAPs can be exploited are:

⁵<https://www.figure-eight.com/>

⁶www.mturk.com

2.3. COLLECTIVE INTELLIGENCE, HUMAN COMPUTATION AND CROWDSOURCING

- image annotation (e.g ESP game [27])
- object localization in images (e.g Peakaboom [28])
- document labelling (e.g GeAnn [29])
- text processing summarization (e.g Verbosity [30])
- web search improvement (e.g. PageHunt [31])
- medical tasks (e.g. Foldit [32])

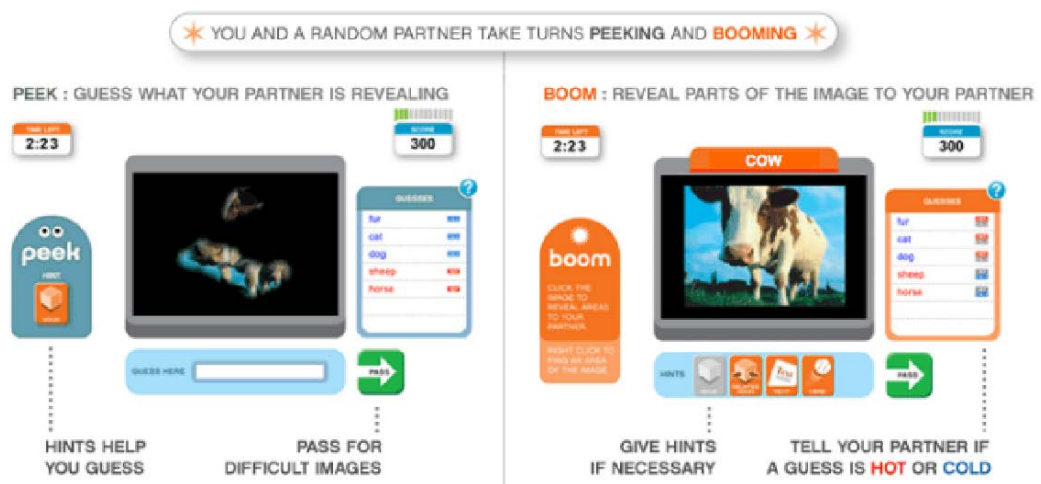


Figure 2.4: Peakaboom Game interface

2.4 Crowdsourcing in IR

In the last years, Crowdsourcing has begun to be applied to Information retrieval tasks: most research has focused on finding strategies for reducing the time, cost and effort of the work on tasks actually done by company workers. Such tasks include the annotation of documents to be used, for example, in learning to rank algorithms [33], relevance assessment of documents for ground truth creation and other manual tasks necessary to IR systems testing and usage.

Crowdsourcing has also been used to validate search results: in [34], Yan and Kumar proposed an image search engine for mobile phones where questionable data is validated on Mturk by crowd workers.

In the following, we describe how crowdsourcing is leveraged for relevance evaluation and how can spam workers be detected.

2.4.1 Crowdsourcing for relevance evaluation

Before crowdsourcing development, the only available test collections were those created in evaluation campaigns following the Cranfield paradigm already explained in section 2.2.1. New applications and studies, however, present some new needs that have pushed towards a different way of creating such collections:

- Fast implementation: the classic development process might be too long
- Low-cost collections: the classic process might be too costly to be performed by a single company
- Domain-specific collections: Standard collections might not be useful to test some systems
- Collection extension: during the process, may be necessary to increase the amount of data to be used in experiments

In order to compute collections in a crowdsourced way, every topic-document pair is then given to a set of crowd workers (in the form of a HIT on crowdsourcing platforms), and each of them supplies his own relevance judgement for the document on the given topic. In Figure 2.5 are represented an example

of HIT on Mturk, and its XML representation.

Relevance Evaluation

Instructions

Please evaluate the relevance of the following text fragment.

Is the following text relevant to **Andorra**?

Tourism, the mainstay of Andorra's tiny, well-to-do economy, accounts for more than 80% of GDP. An estimated 11.6 million tourists visit annually, attracted by Andorra's duty-free status and by its summer and winter resorts.

☐ Irrelevant
☐ Marginally relevant
☐ Fairly relevant
☐ Highly relevant

```

<Question>
  <QuestionIdentifier>question1</QuestionIdentifier>
  <DisplayName>Question 1:</DisplayName>
  <IsRequired>true</IsRequired>
  <QuestionContent>
    <FormattedContent><![CDATA[
      Is the following text relevant to Andorra?
      Tourism, the mainstay of Andorra's tiny, well-to-do economy, accounts for more than 80%
      of GDP. An estimated 11.6 million tourists visit annually, attracted by Andorra's duty-free
      status and by its summer and winter resorts.
    ]]></FormattedContent>
  </QuestionContent>
  <AnswerSpecification>
    <SelectionAnswer>
      <StyleSuggestion>radiobutton</StyleSuggestion>
      <Selections>
        <Selection>
          <SelectionIdentifier>ir</SelectionIdentifier>
          <Text>Irrelevant</Text>
        </Selection>
        <Selection>
          <SelectionIdentifier>mr</SelectionIdentifier>
          <Text>Marginally relevant</Text>
        </Selection>
        <Selection>
          <SelectionIdentifier>fr</SelectionIdentifier>
          <Text>Fairly relevant</Text>
        </Selection>
        <Selection>
          <SelectionIdentifier>hr</SelectionIdentifier>
          <Text>Highly relevant</Text>
        </Selection>
      </Selections>
    </SelectionAnswer>
  </AnswerSpecification>
</Question>
  
```

Figure 2.5: Relevance evaluation task on MTurk

All the relevance judgements are then used, in place of the experts' ground truth, to evaluate IR systems' performance.

A lot of studies [35–37] have been made to inspect the difference between crowd judgements and expert judgements: the strong assumption on which most of these studies rely on is that experts' judgement is the gold standard, meaning that experts' judgement is the best judgement crowd workers can reach. Even if this assumption is far to be proved [38], judgements provided by crowd assessors often agree with experts' judgements and, in particular, crowd workers tend to agree a bit more with the experts when the document is

relevant, and less when it is not relevant [35, 39], highlighting a limit of crowd workers in detecting poor document performance.

Some studies analyse how different levels of expertise [40–42], nationality and remuneration methods [43] of crowd assessors can lead to different accuracy in results.

A more general study [44] tries to investigate what disagreement means, saying that there’s no guarantee that all disagreement is due to workers’ ineffectiveness. In figure 2.6 we report the possible reasons for disagreement described in [44].

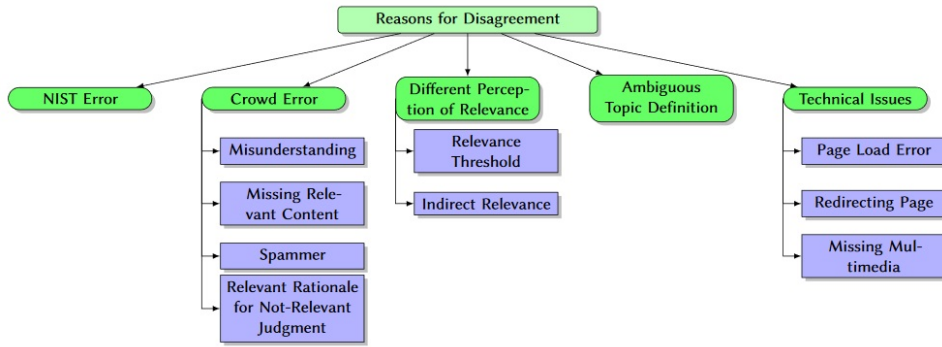


Figure 2.6: Reasons for Crowd-expert disagreement

2.4.2 Noisy judgements

Connecting to what has just been said, we address now the main problem that must be taken into consideration, reporting the main methods applied to solve it: since most of the crowd assessors are motivated by monetary reward, a non-negligible part of them [45] provide inaccurate judgements maximizing earnings to the detriment of work quality.

Spam judgements can be classified [46] into:

- Sloppy workers, whose mistakes in judgement are honest ones, without the intent to provide bad results. Possible causes of mistake can be both the lack of clarity in the instructions of the HIT or a poor worker competence.
- Random Spammers, that purposely randomize their judgements.
- Uniform Spammers, that use a fixed pattern in their judgements.

Even if mechanisms described in section 2.3 can be used to reduce spam at the source, to further improve crowd judgements accuracy two main strategies are applied: spam filtering and assessor weighting.

Spam filtering is about removing spam assessors' judgements. This approach has been followed by Vuurens and De Vries in [46]: they provide two different mechanisms to detect random spammers and uniform spammers. Random spammers are detected computing the average squared distance in relevance labels between each assessor's judgements and the judgements of all other assessors. Uniform spammers are detected counting the averaged squared number of disagreements that each worker has with other workers while repeating voting patterns. Other spam-detection algorithms are developed using Machine Learning techniques, as reported in [47].

The risk to be taken while filtering assessors, is to falsely reject workers that don't agree with the majority vote [46].

Another way to take into account the different quality of assessors' judgements is assigning a weight to each assessor, used to aggregate judgements in a more efficient way.

The work of this thesis follow this school of thought: in the next section will be described some methods for aggregating crowd judgements, some of them will be compared with the thesis work.

2.5 Crowdsourcing techniques

In this section, we present some of the state-of-the-art approaches for exploiting crowd judgements in place of expert relevance judgements in retrieval evaluation.

The classic approach to the problem is to create a merged ground truth from assessors' judgements, to be used in place of expert ground truth. The most common techniques in this category are Majority vote (Section 2.5.1), with its weighted variants [6, 48, 49], and Expectation maximization (Section 2.5.2), but several other approaches can be found in literature. In TurkRank (Section 2.5.4) the ground truth is estimated through an algorithm working on a graph representation of the assessors' judgements, in GeAnn game (Section 2.5.3), ground truth is estimated merging crowd judgements on small portions of the documents' corpus, in Skierarchy (Section 2.5.5) machine learning, crowd

assessors and experts are used together to compute quality judgements.

A different approach to the problem is to independently use the judgements given by every single crowd assessor to evaluate IRS effectiveness, merging the assessors' data at measure level. In AWARE probabilistic framework [9] this approach is followed weighting assessors according to accuracies computed in an unsupervised way.

S-Aware, subject of this thesis, is proposed as a new component of the AWARE framework, based on supervised methods for estimating assessors' accuracies. Aware framework will be the subject of the next chapter.

2.5.1 Majority vote

The most intuitive way to achieve a ground truth based on crowd assessors' judgement is to use the Majority Vote (MV) algorithm: for every topic-document pair, the ground truth judgement is set to the judgement given by the majority of crowd assessors.

Formally, let D and T be respectively a set of documents and a set of topics, let (REL, \leq) be a totally ordered set of relevance degrees, let $\Lambda = \{W_1, \dots, W_k, \dots, W_l\}$ be a set of workers and let $GT_k(t, d) \in \{0, 1\}$ be the relevance judgement given by assessor k for topic t on document d . The ground truth value for (t, d) pair can be computed as:

$$GT(t, d) = \underset{g \in REL}{argmax} \sum_{k=1}^l \mathbb{1}_{\{GT_k(t, d)=g\}}$$

where $\mathbb{1}_{\{GT_k(t, d)=g\}}$ is equal to 1 if assessor k judged document d as relevant for topic t with relevance grade g , 0 otherwise.

Despite its simplicity, this algorithm works quite well, but does not take into account the possible different levels of expertise among assessors.

To overcome this problem, several weighted versions of majority vote has been developed [6, 48, 49].

The idea behind this weighted algorithms is that judgements from crowd assessors must be weighted by a coefficient representing the assessor accuracy. If w_k is the accuracy coefficient assigned to assessor k , the ground truth value for (t, d) pair can be expressed as:

$$GT(t, d) = \underset{g \in REL}{argmax} \sum_{k=1}^l w_k \mathbb{1}_{\{GT_k(t, d)=g\}}$$

Algorithms [48, 49] differ in how this accuracy coefficients are computed.

In [49] the problem of compute weights is treated as a machine learning domain adaptation problem, modelling each assessor’s knowledge with a labelling function: weights w_k are computed minimizing

$$\|g - w_k f_k\|_2^2$$

where f_k is the labelling function obtained training on relevance judgements given by assessor k and g is an estimate of the gold labelling function obtained training on all crowd assessors’ judgements.

In [49] a second, more advanced, version of the algorithm is then presented, trying to use labelling functions that model assessor features.

In [48] weights are estimated in an iterative way, with an algorithm that consists of four main steps, where steps 2, 3 and 4 are looped until results converges:

1. Initialization: Set uniform weights for assessors
2. Loop: Estimate current gold relevance judgements with weighted MV algorithm.
3. Loop: Compute assessor weight as

$$w_k = \frac{\# \text{agreements between current gold judgements and assessor } k}{\# \text{documents judged by assessor } k}$$

4. Loop: Emphasize weights computing

$$w_k = 2w_k - 1$$

The operation on step 4 is done in order to give higher weight to good assessors, rapidly downplaying spammers (spammers’ opinion is less considered).

For completeness, we present a probabilistic version of the MV algorithm, presented in [50]. We consider each assessor’s judgement as a Binomial random variable and we assume such variables to be i.i.d. Then, the ground truth value for a given topic-document pair can also be modelled as a Binomial variable of parameter:

$$p_{t,d} = \frac{1}{M} \sum_{k=1}^M AS_k(t,d)$$

where M is the number of assessors and $AS_k(t, d)$ is the generic random variable modelling the relevance judgement of assessor k on topic t for document d .

2.5.2 Expectation maximization

Expectation Maximization (EM) algorithm [2] concurrently estimates documents' relevance and workers' accuracy until convergence.

As described in [2, 9], we define $p_t[g] = \mathbb{P}[GT(t, \cdot) = g]$ as the probability that a given document has relevance grade g and we define a latent confusion matrix $\pi_t[\cdot, \cdot](k)$ for each assessor k , representing t topic assessor's judgements probability based on the true ones:

$$\pi_t[g, h](k) = \mathbb{P}[GT_k(t, \cdot) = h \mid GT(t, \cdot) = g]$$

is the probability that the assessor k provides relevance grade h for document d , given that document d has g as true relevance judgement.

An estimate of such value can be computed as:

$$\pi_t[g, h](k) = \frac{\text{\#assessor judgements with grade } h \text{ when true grade is } g}{\text{\#total assessor judgements when true grade is } g}$$

The EM algorithm consists of five steps:

1. Initialize $p_t[g]$ and $\pi_t[\cdot, \cdot](\cdot)$
2. Compute Maximum likelihood estimates of $p_t[g]$ and $\pi_t[\cdot, \cdot](\cdot)$

$$\tilde{\pi}_t[g, h](k) = \frac{\sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t, d)=g\}} \mathbb{1}_{\{GT_k(t, d)=h\}}}{\sum_{h \in REL} \sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t, d)=g\}} \mathbb{1}_{\{GT_k(t, d)=h\}}}$$

$$\tilde{p}_t[g] = \frac{\sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t, d)=g\}}}{|D|}$$

Where $\mathbb{1}_{\{GT(t, d)=g\}} \mathbb{1}_{\{GT_k(t, d)=h\}}$ is equal to 1 if and only if assessor k judged the document d as relevant with relevance grade h , given that the correct relevance grade is g .

3. Compute the new estimate of the ground truth with:

$$\begin{aligned} & \mathbb{P}[GT(t, d) = g \mid GT.(t, \cdot), \pi_t[\cdot, \cdot](\cdot)] = \\ & = \frac{\tilde{p}_t[g] \prod_{k=1}^m \prod_{h \in REL} (\tilde{\pi}_t[g, h](k))^{\mathbb{1}_{\{GT_k(t, d)=h\}}}}{\sum_{g \in REL} \tilde{p}_t[g] \prod_{k=1}^m \prod_{h \in REL} (\tilde{\pi}_t[g, h](k))^{\mathbb{1}_{\{GT_k(t, d)=h\}}}} \end{aligned}$$

4. Repeat steps 2 and 3 until the results convergence
5. Define relevance labels: for each document label g is assigned if g is the label with maximum probability of relevance. In binary case, documents are set to relevant if $p_t[1] \geq 0.5$.

Since EM algorithm finds a local optimum value for relevance probabilities and latent matrices, a crucial point is defining the initial values for these variables. Several different strategies can be found in literature:

- random initialization [2]
- MV seeding: MV algorithm is used to find the initial ground truth relevance labels [3]
- Semi supervised approach: a small set of expert labels is used together with MV estimated labels[5]
- Assessors' honesty hypothesis: variables are initialized assuming that assessors are honestly attempting to give correct answers, so elements in the principal diagonal of latent confusion matrices are initialized to a high value ($\pi_t[\alpha, \alpha](k) = 0.9$) [4]

2.5.3 GeAnn

GeAnn is a term association game proposed at the TREC 2011 crowdsourcing Track [29]. The idea behind this game is to collect relevance judgements from the crowd using a game with a purpose.

Instead of collecting judgements on documents as a whole, the judgement process is broken down onto term level. In Figure 2.7 is represented the game interface: each document sentences are taken separately, and user goal is to match the main keyword of the sentence with the correct bucket. In a postprocessing step, sentence-level judgements are aggregated in two different ways, using a MV algorithm or a weighted MV using a reliability function based on assessor agreement with some gold judgements.

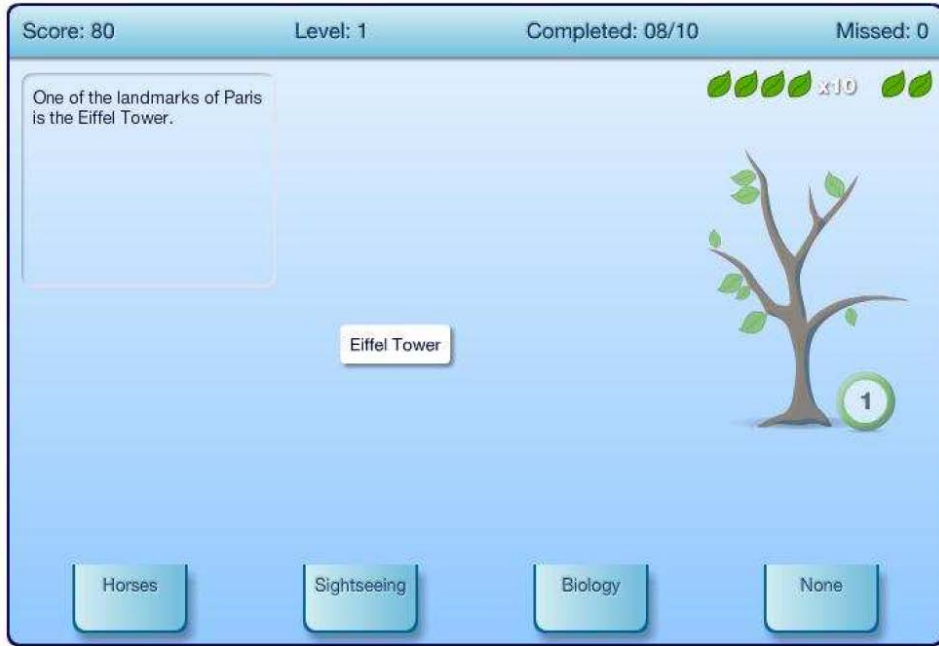


Figure 2.7: screen view of the geAnn game

2.5.4 TurkRank

TurkRank [7] is a network based approach for detecting assessors' trustworthiness. Trustworthiness is incrementally updated computing both crowd assessors inter-agreement and gold standard agreement: the more an assessor agrees with the others, the greater is his trustworthiness.

In Figure 2.8 is represented an example of network model, nodes represent assessors (both crowd and gold) and edges represent agreement between assessors.

TurkRank is implemented using a modified version of the PageRank with Priors (PRwP) algorithm [51], where assessors take the place of web pages.

Trustworthiness is computed as:

$$\pi(v)^{i+1} = (1 - \beta) \left(\sum_{u=1}^{d_{in}(v)} p(v | u) \pi^{(i)}(u) \right) + \beta p_v$$

where $0 \leq \beta \leq 1$ is a parameter, and $u = 1, \dots, d_{in}(v)$ are the assessors that agree with v . To the gold standard vertex is assigned a $p_v = 1$ prior probability, setting $p_v = 0$ for all other nodes, so if $\beta=1$ only gold standard assessor can accumulate

trust, if $0 \leq \beta \leq 1$ assessors' agreement and gold standard agreement will be combined.

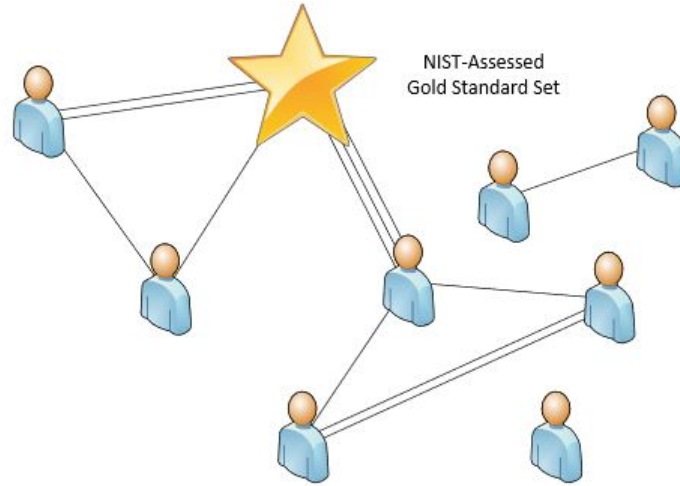


Figure 2.8: Example of network model, showing crowd assessors agreements between crowd assessors and with gold standard

2.5.5 Skierarchy

Skierarchy [8] is a hierarchical approach developed by SetuServ inc. to test how crowdsourcing can be used in domain specific data analytic tasks, that are currently managed only by domain experts.

The base idea is that experts' judgement cannot be eliminated, but can be significantly reduced if used together to other types of assessments: a small number of domain experts are used to train and supervise a large set of crowd assessors. Experts break down the complex tasks into crowdsourceable microtasks, train and supervise crowd assessors while performing microtasks and solve difficult microtask that crowd assessors are not able to solve.

In order to improve crowd assessor performance, a machine learning algorithm is exploited to predict scores for the microtasks and to create annotation suggestion to help the assessor in the assessing process (Figure 2.9).

The steps of Skierarchy process are:

- Crowd training: crowd workers are asked to evaluate a set of documents. Experts then examine the results and explain to crowd assessors their

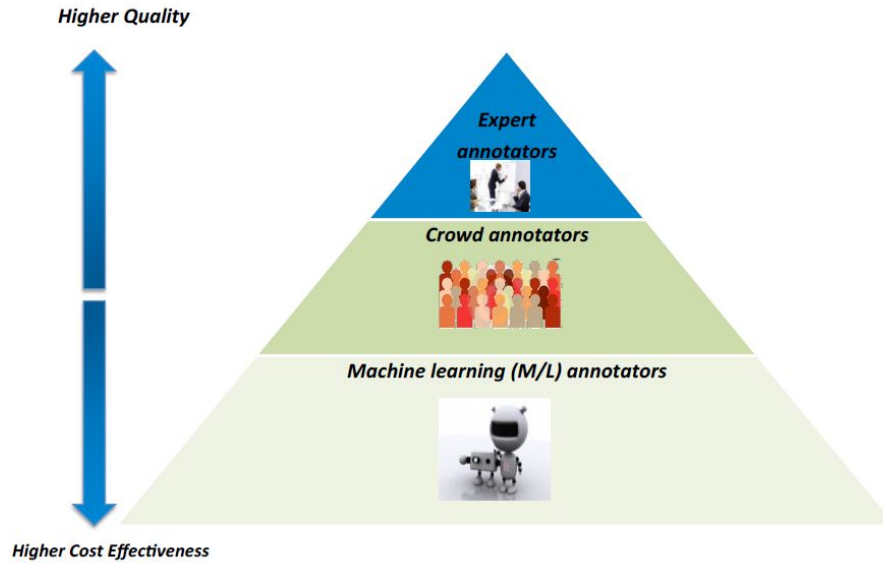


Figure 2.9: Hierarchy of assessors used in Skierarchy

errors in assessing documents.

- **Machine learning Model:** the judgements are then used as training set for a ML algorithm that uses logistic regression to classify all the remaining documents in different classes, computing a relevance score for each document.
- **Crowd annotation:** the documents are divided in buckets based on the ML algorithm scores, each bucket is assigned to a crowd assessor. Buckets with higher scores are assigned to the assessors that better performed in the training phase, because this documents are like to be relevant and we aim to avoid misses.
- **Automatic error correction:** ML algorithm is retrained using the complete annotated dataset, and is used to compute relevance scores on the documents using 10 fold validation. Documents for which algorithm and assessor disagree are then revised by the assessors to determine which is the correct judgement.

AWARE FRAMEWORK

The work of this thesis is based on the work presented in [9], that addresses the problem of ground truth creation from a different point of view with respect to classical approaches described in the previous chapter.

AWARE (Assessor-driven Weighted Averages for Retrieval Evaluation) [9] probabilistic framework, differently from all other approaches, allows to combine assessors' knowledge at measure level, instead of combining judgements at pool level: Figure 3.1 represents this methodology.

The main idea that motivates this decision is that aggregation intrinsically implies loss of information, and then postponing the aggregation process can lead to a more accurate measure computation.

Even small errors in merged ground truth are in fact propagated while computing evaluation measures, and the same error at pool level can have a different impact on different measures or systems.

The AWARE framework describes different ways in which the evaluation measures based on the ground truth generated by each assessor can be merged in a single measure, called the AWARE version of the measure.

In order to formally define how this merged measures are computed, we define the judged run for assessor k as the function which assigns a relevance degree to each retrieved document in a run.

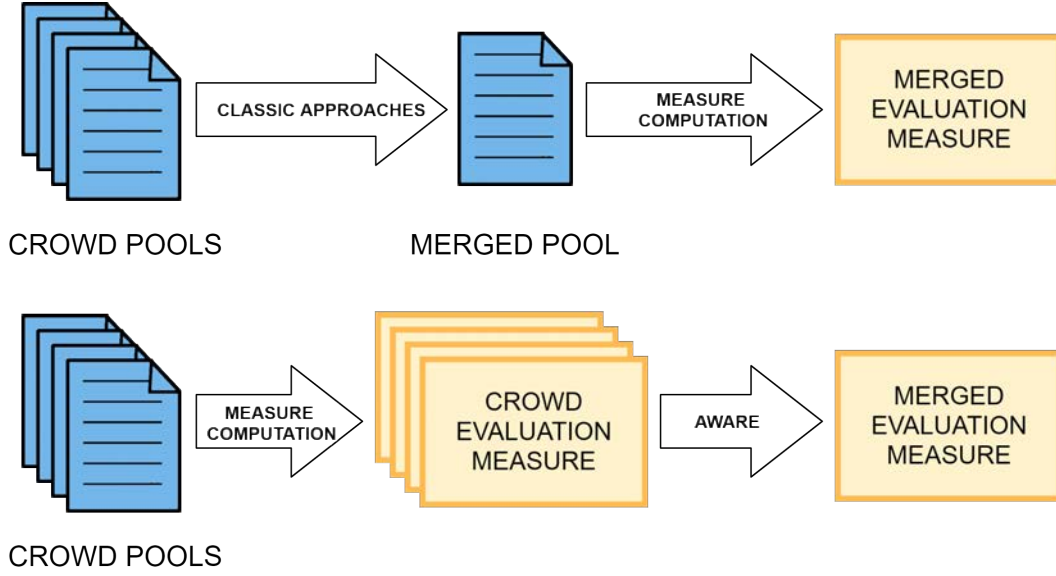


Figure 3.1: Classic approaches methodology vs AWARE methodology

$$(t, r_t, k) \rightarrow \hat{r}_t^k = (GT^k(t, d_1), GT^k(t, d_2), \dots, GT^k(t, d_n))$$

The simplest way to aggregate crowd assessors' measures is to assume that all the assessors are equally responsible for relevance evaluation: in this case, measures are aggregated giving the same importance to the measures from each assessor.

This is the so called uniform AWARE version of the measure, defined as follows:

$$aware - m(t, r_t)_{uni} = \frac{1}{m} \sum_{k=1}^m \mu(\hat{r}_t^k)$$

where m is the number of merged crowd assessors, and $\mu(\hat{r}_t^k)$ is the evaluation measure computed on run r_t according to k -th assessor judgements.

This methodology might already be sufficient to improve measure computation: we report a simple example [9] of the comparison between uni-AWARE approach and the classic Majority vote approach.

Lets consider a pool containing 3 relevant documents and run of 5 documents where first and third documents are relevant

$$\hat{r}_t = (1, 0, 1, 0, 0)$$

Average precision for this run can be computed as described in section 2.2.3

$$AP(\hat{r}_t) = \frac{\frac{1}{1} + \frac{2}{3}}{3} = 0.5556$$

Let three assessors evaluate the documents in the run, giving different judgements.

$$\hat{r}_t^1 = (1, 1, 0, 0, 0) \quad \hat{r}_t^2 = (1, 1, 1, 0, 0) \quad \hat{r}_t^3 = (0, 1, 1, 0, 1)$$

With MV approach, we can compute a merged ground truth and then we can compute AP with respect to the MV pool:

$$\hat{r}_t^{MV} = (1, 1, 1, 0, 0) \rightarrow AP(\hat{r}_t^{MV}) = \frac{\frac{1}{1} + \frac{2}{2} + \frac{3}{3}}{3} = 1.00$$

which represents a 80% error with respect to the gold version of the measure. With AWARE approach we compute AP using the single assessors' judgements and merge them into the aware version of the measure:

$$AP(\hat{r}_t^1) = \frac{\frac{1}{1} + \frac{2}{2}}{3} = 0.67 \quad AP(\hat{r}_t^2) = \frac{\frac{1}{1} + \frac{2}{2} + \frac{3}{3}}{3} = 1.00 \quad AP(\hat{r}_t^3) = \frac{\frac{1}{2} + \frac{2}{3} + \frac{3}{5}}{3} = 0.59$$

$$aware - AP(\hat{r}_t)_{uni} = \frac{1}{3} (AP(\hat{r}_t^1) + AP(\hat{r}_t^2) + AP(\hat{r}_t^3)) = 0.75$$

which represents only a 35% error with respect to the gold version of the measure.

Motivated from this results, and from the belief that real crowd assessors are far to be uniformly experienced on the analysed topics, we move to a weighted version of the AWARE measure, in which each assessor is weighted by a score representing its judgement accuracy:

$$aware - m(t, r_t) = \sum_{k=1}^m \mu(\hat{r}_t^k) a_k(t)$$

where accuracies $a_k(t)$ can be computed at an overall level or topic by topic.

AWARE framework provides a wide range of accuracy estimators for crowd assessors accuracy, that can be divided into two main classes:

- u-AWARE approaches, presented in [9] and described in section 3.2, provide unsupervised estimators for accuracy scores based on comparisons between crowd assessors and random assessors.
- s-AWARE approaches: original contribute of this thesis described in section 3.3, provide supervised estimators for accuracy scores based on

comparisons between crowd assessors and the gold standard on some training topics.

Comparisons in both u-AWARE and s-AWARE methods use some dissimilarity measures between assessors measures.

Since it's not clear how the dissimilarity between two assessors can be correctly computed, different types of measures are considered. In section 3.1 we describe the different dissimilarity measures used in AWARE framework estimators.

3.1 Dissimilarity definitions

In order to explore a wide range of ways in which an evaluation measure can be different from another, we analyze three categories of dissimilarities:

- **Measure dissimilarity:** the dissimilarity is computed using directly the values of the measures
- **Distribution dissimilarity:** the dissimilarity is computed using the probability distributions of the measures, in particular we measure how much different is each assessors' distribution with respect to the gold standard distribution
- **Ranking dissimilarity:** systems are sorted by measure value and dissimilarity is computed in term of ranking comparisons

3.1.1 Measure dissimilarity

Frobenius Norm Given an $m \times n$ matrix A , its Frobenius Norm is

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

Using Frobenius norm, the dissimilarity between two measure matrices can be computed as $\|M_1 - M_2\|_F$

Root Mean Squared Error Given an two vectors X and Y , their RMSE

$$RMSE = \sqrt{\sum_{i=1}^m \frac{(X_i - Y_i)^2}{m}}$$

3.1.2 Distribution dissimilarity

Kullback-Leibler Divergence Kullback-Leibler Divergence is a measure of how a probability distribution Y is different from a reference distribution X . In order to use KLD for our purposes, we must estimate the probability distribution (PDF) of the assessors' performance measures. To do this, we use Kernel density estimation (KDE): given a vector X of m elements, the KDE estimation of its PDF is:

$$\hat{f}_X(x) = \frac{1}{mb} \sum_{i=1}^m K\left(\frac{x - X_i}{b}\right)$$

where K is a function satisfying $\int_{-\infty}^{\infty} K(x)dx = 1$ and b is a smoothing function called bandwidth. Once computed PDFs, Kullback-Leibler Divergence is given by:

$$D_{KL}(X \parallel Y) = \sum_x \ln \left(\frac{\hat{f}_X(x)}{\hat{f}_Y(y)} \right) \hat{f}_X(x)$$

3.1.3 Ranking dissimilarity

Kendall Tau Correlation Considering two vectors X and Y of m elements, we can define their Kendall's τ correlation as

$$\tau(X, Y) = \frac{C - D}{m(m-1)/2}$$

where C is the number of pairs ranked in the same order in X and Y , and D is the number of discordant pairs.

AP correlation AP correlation is a top heavy measure inspired by Kendall's Tau that gives more importance to top ranked elements.

Considering two vectors X and Y of m elements, we can define their AP correlation as

$$\tau_{AP}(Y, X) = \frac{2}{(m-1)} \sum_{i=2}^m \frac{C(i)}{i-1} - 1$$

where $C(i)$ is the number of items above rank i in X which are ranked above $x[i]$ in Y .

3.2 u-AWARE accuracy computation

u-AWARE part of the framework [9] describes several methods to merge together performance measures from multiple assessors based on unsupervised estimators for crowd assessors' accuracies.

To evaluate accuracy, we compute the dissimilarity between evaluation measures based on the crowd assessor's judgements and measures based on judgements coming from different types of random assessors': the greater the dissimilarity, the better the crowd assessor's accuracy.

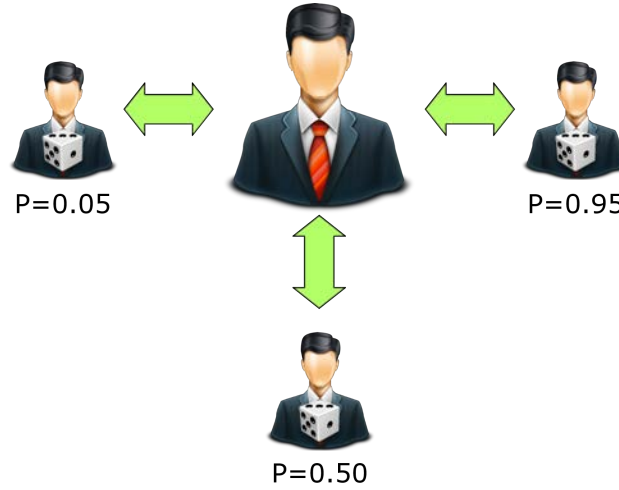


Figure 3.2: u-AWARE accuracy is computed with dissimilarity measures between crowd assessors and different types of random assessors

A first classification between accuracy scores can be done looking at granularity: we can compute a score for each topic (then we have topic-by-topic granularity - tpc) or a single score for all the topics (then we have single score granularity - sgl).

In order to compute both sgl and tpc accuracy scores, two main steps are followed:

- **GAP computation:** Evaluation measures are computed taking as ground truth the judgements given by each crowd assessor. The same evaluation measures are computed taking as ground truth the judgements given by three classes of random assessors, which randomly assess documents

with different probability of relevance ($p = 0.05; p = 0.50; p = 0.95$). GAP is then computed between assessor measures and random measures (Figure 3.2).

- Weight computation: the three GAP dissimilarities are aggregated to compute accuracy

In the next paragraphs five different GAP approaches and three weight approaches will be described in detail. In Figure 3.3 are shown the 15 combinations of GAP and weight approaches tested in [9]. AWARE algorithms for sgl and tpc accuracy weights are shown in Algorithms 1 and 2.



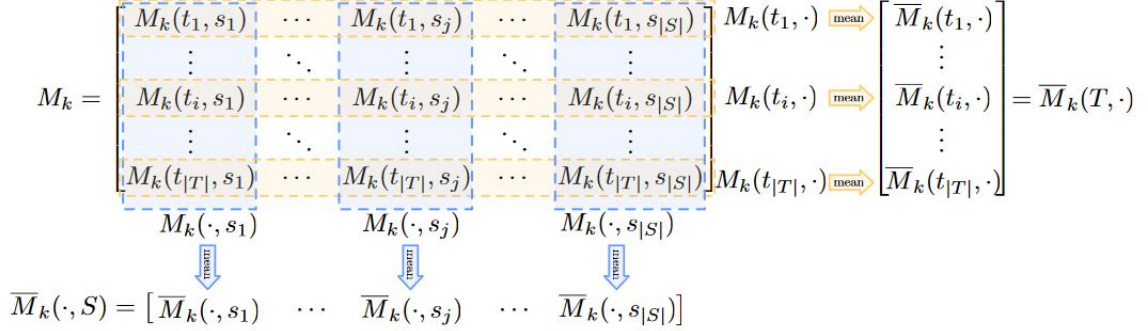
Measure	Gap G_k	Weight w_k		
		Minimal Dissimilarity	Minimal Squared Dissimilarity	Minimal Equi Dissimilarity
 M_h^p Random Assessors ρ_h^p	Measure Level - Frobenius Norm - RMSE	fro_md rmse_md	fro_msd rmse_msd	fro_med rmse_med
 M_k Crowd Assessor W_k	Distribution Level - KL Divergence	kld_md	kld_msd	kld_med
	Rankings Level - Kendall's Tau - AP Correlation	tau_md apc_md	tau_msd apc_msd	tau_med apc_med

Figure 3.3: AWARE GAP-Weight combinations

3.2.1 GAP

Let S be a set of IR systems and let T be a set of topics: we define a $|T| \times |S|$ matrix containing the values of a performance measure computed on the runs generated by each system for each topic. Figure 3.4 represent such matrix and some notation: $\overline{M}_k(\cdot, s)$ and $\overline{M}_k(t, \cdot)$ are respectively the marginal mean across the rows and across the columns.

According to definition given in section 3.1, we describe how each measure is exploited to compute unsupervised GAPs.


 Figure 3.4: $|T| \times |S|$ matrix for the assessor measures

While describing GAP computation, we describe also how each GAP is normalized to obtain comparable scores: each GAP measure G' is normalized in the $[0,1]$ range, where $G' = 0$ means that assessor follows the behaviour of the random assessor, $G' = 1$ means that assessor k is far from being a random assessor.

Frobenius Norm Frobenius norm GAP is computed as the norm of the difference between assessor's measures and random measures

$$G_k^p = \|M_k - M_h^p\|_F \quad G_k^p(t) = \|M_k(t, \cdot) - M_h^p(t, \cdot)\|_F$$

where M_h^p represent measures from h -th random assessor assessing relevant documents with probability p .

To obtain values in the range $[0,1]$, the following normalization is applied:

$$G' = \frac{G}{\sqrt{|T| \cdot |S|}}$$

Root Mean Squared Error RMSE GAP is computed as the RMSE between assessor's measures and random measures. In the sgl case, measures are averaged by topic before RMSE computation.

$$G_k^p = RMSE(\overline{M}_k(\cdot, S) - \overline{M}_h^p(\cdot, S)) \quad G_k^p(t) = RMSE(M_k(t, \cdot) - M_h^p(t, \cdot))$$

Since evaluation measures take value in the $[0,1]$ range, then also RMSE takes values in that range. RMSE=0 means no difference from random assessor, so no normalization is required.

$$G' = G$$

Kullback-Leibler Divergence KLD GAP is computed as the KLD between the PDF of assessor measures and the PDF of random measures.

$$G_k^p = D_{KL}(M_k(\cdot, \cdot) \parallel M_h^p(\cdot, \cdot)) \quad G_k^p(t) = D_{KL}(M_k(t, \cdot) \parallel M_h^p(t, \cdot))$$

KLD takes values in $[0, +\infty)$ in order to obtain values in the range $[0, 1]$, the following normalization is applied:

$$G' = 1 - e^{-\beta G}$$

where β is a positive real number.

Kendall Tau Correlation Tau correlation GAP is computed as the correlation between the system rankings induced respectively by crowd and random assessors' measures.

$$G_k^p = \tau(\overline{M}_k(\cdot, S), \overline{M}_h^p(\cdot, S)) \quad G_k^p(t) = \tau(M_k(t, \cdot), M_h^p(t, \cdot))$$

High correlation means small dissimilarity from random assessors, and then a poor performance. To obtain values in the range $[0, 1]$, the following normalization is applied:

$$G' = 1 - |G|$$

AP correlation AP correlation GAP is computed as the correlation between the system rankings induced respectively by crowd and random assessors' measures.

$$G_k^p = \tau_{AP}(\overline{M}_k(\cdot, S), \overline{M}_h^p(\cdot, S)) \quad G_k^p(t) = \tau_{AP}(M_k(t, \cdot), M_h^p(t, \cdot))$$

To obtain values in the range $[0, 1]$, we apply the same normalization of Kendall tau GAP:

$$G' = 1 - |G|$$

3.2.2 Weight

The three ways we use to aggregate GAPs from random assessors are:

- **Minimal Dissimilarity:** the accuracy weight is computed as the minimum GAP from the tree random assessor classes. In this case we consider the value for which is impossible to the crowd assessor to be closer to another random assessor.

$$a_k = \min\left((G_k^{0.05})', (G_k^{0.50})', (G_k^{0.95})'\right)$$

- **Minimal Squared Dissimilarity:** the accuracy weight is computed as the minimum squared GAP from the tree random assessor classes. In this case we reason as in the previous case, but we aim to emphasize small GAPs.

$$a_k = \min \left(\left((G_k^{0.05})' \right)^2, \left((G_k^{0.50})' \right)^2, \left((G_k^{0.95})' \right)^2 \right)$$

- **Minimal Equi-Dissimilarity:** the accuracy weight is computed as the sum of the three GAPs. Here we state that good assessors have to behave different from all three random behaviour under consideration.

$$a_k = (G_k^{0.05})' + (G_k^{0.50})' + (G_k^{0.95})'$$

Algorithm 1: How to compute sgl accuracy with u-AWARE

Data: T set of topics; $p \in \{0.05, 0.50, 0.95\}$ probability of relevance for random assessor judgements; $H \in \mathbb{N}$ number of random assessors replicates $\forall p$; $\hat{r}_t^k \forall t \in T$ ground truth generated by assessor k ; $\hat{r}_{t,h}^p \forall t \in T$ ground truth generated by the h -th random assessor of level p

Result: a_k sgl accuracy score for k -th assessor

- 1 /* Compute the measures M_k for the k -th assessor and M_h^p for each random assessor */;
- 2 $M_k \leftarrow$ compute $m(\cdot)$ on \hat{r}_t^k ;
- 3 $M_h^p \leftarrow$ compute $m(\cdot)$ on $\hat{r}_{t,h}^p \forall h = \{1, \dots, H\}$ and $\forall p \in \{0.05, 0.50, 0.95\}$;
- 4 /* compute and normalize GAP $G_{k,h}^p$ with respect to each random assessor $\forall h = \{1, \dots, H\}$ and $\forall p \in \{0.05, 0.50, 0.95\}$ */
- 5 **for** $h \in 1, \dots, H$ **do**
- 6 **if** *frobenius norm* **then**
- 7 $G_{k,h}^p = \|M_k - M_h^p\|_F \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* GAP computation;
- 8 $(G_{k,h}^p)' = \frac{G_{k,h}^p}{\sqrt{|T| \cdot |S|}} \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* [0,1] normalization;
- 9 **else if** *RMSE* **then**
- 10 $G_{k,h}^p = RMSE(\overline{M}_k(\cdot, S) - \overline{M}_h^p(\cdot, S))$ /* GAP computation;
- 11 $(G_{k,h}^p)' = G_{k,h}^p \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* [0,1] normalization;
- 12 **else if** *KL divergence* **then**
- 13 $G_{k,h}^p = D_{KL}(M_k(\cdot, \cdot) \| M_h^p(\cdot, \cdot))$ /* GAP computation;
- 14 $(G_{k,h}^p)' = 1 - e^{-\beta G_{k,h}^p} \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* [0,1] normalization;
- 15 **else if** *Kendall Tau* **then**
- 16 $G_{k,h}^p = \tau(\overline{M}_k(\cdot, S), \overline{M}_h^p(\cdot, S))$ /* GAP computation;
- 17 $(G_{k,h}^p)' = 1 - |G_{k,h}^p| \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* [0,1] normalization;
- 18 **else if** *AP correlation* **then**
- 19 $G_{k,h}^p = \tau_{AP}(\overline{M}_k(\cdot, S), \overline{M}_h^p(\cdot, S))$ /* GAP computation;
- 20 $(G_{k,h}^p)' = 1 - |G_{k,h}^p| \quad \forall p \in \{0.05, 0.50, 0.95\}$ /* [0,1] normalization;
- 21 /* Aggregate the GAP with respect to the random assessors replicates */
- 22 $(G_k^p)' \leftarrow \text{mean}((G_{k,h}^p)')$
- 23 /* compute assessor accuracy weight */
- 24 **if** *minimal dissimilarity* **then**
- 25 $a_k = \min((G_k^{0.05})', (G_k^{0.50})', (G_k^{0.95})')$
- 26 **else if** *minimal squared dissimilarity* **then**
- 27 $a_k = \min(((G_k^{0.05})')^2, ((G_k^{0.50})')^2, ((G_k^{0.95})')^2)$
- 28 **else if** *minimal equi-dissimilarity* **then**
- 29 $a_k = (G_k^{0.05})' + (G_k^{0.50})' + (G_k^{0.95})'$

Algorithm 2: How to compute tpc accuracy with u-AWARE

Data: T set of topics; $H \in \mathbb{N}$ number of random assessors replicates $\forall p \in \{0.05, 0.50, 0.95\}$; \hat{r}_t^k
 $\forall t \in T$ ground truth generated by assessor k ; $\hat{r}_{t,h}^p$ $\forall t \in T$ ground truth generated by the
 h -th random assessor of level p

Result: a_k vector of $|T|$ elements containing tpc accuracy scores for k -th assessor

- 1 /* Compute the performance measures M_k for the k -th assessor and M_h^p for each random
 assessor */;
- 2 $M_k \leftarrow$ compute $m(\cdot)$ on \hat{r}_t^k ;
- 3 $M_h^p \leftarrow$ compute $m(\cdot)$ on $\hat{r}_{t,h}^p$ $\forall h = \{1, \dots, H\}$ and $\forall p \in \{0.05, 0.50, 0.95\}$;
- 4 /* compute and normalize GAP $G_{k,h}^p(t)$ with respect to each random assessor $\forall h = \{1, \dots, H\}$
 and $\forall p \in \{0.05, 0.50, 0.95\}$ */
- 5 **for** $h \in 1, \dots, H, t \in 1, \dots, T$ **do**
- 6 **if** *frobenius norm* **then**
- 7 $G_{k,h}^p(t) = \|M_k(t, \cdot) - M_h^p(t, \cdot)\|_F$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 8 $(G_{k,h}^p(t))' = \frac{G_{k,h}^p(t)}{\sqrt{|S|}}$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 9 **else if** *RMSE* **then**
- 10 $G_{k,h}^p(t) = \text{RMSE}(M_k(t, \cdot) - M_h^p(t, \cdot))$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 11 $(G_{k,h}^p(t))' = G_{k,h}^p(t)$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 12 **else if** *KL divergence* **then**
- 13 $G_{k,h}^p(t) = D_{KL}(M_k(t, \cdot) \| M_h^p(t, \cdot))$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 14 $(G_{k,h}^p(t))' = 1 - e^{-\beta G_{k,h}^p(t)}$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 15 **else if** *Kendall Tau* **then**
- 16 $G_{k,h}^p(t) = \tau(M_k(t, \cdot), M_h^p(t, \cdot))$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 17 $(G_{k,h}^p(t))' = 1 - |G_{k,h}^p(t)|$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 18 **else if** *AP correlation* **then**
- 19 $G_{k,h}^p(t) = \tau_{AP}(M_k(t, \cdot), M_h^p(t, \cdot))$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 20 $(G_{k,h}^p(t))' = 1 - |G_{k,h}^p(t)|$ $\forall p \in \{0.05, 0.50, 0.95\}$ $\forall p \in \{0.05, 0.50, 0.95\}$;
- 21 /* Aggregate the GAP with respect to the random assessors replicates */
- 22 $(G_k^p(t))' \leftarrow \text{mean}((G_{k,h}^p(t))') \forall p \in \{0.05, 0.50, 0.95\}$ and $\forall t \in \{1, \dots, |T|\}$
- 23 /* compute assessor accuracy weight */
- 24 **for** $t \in \{1, \dots, |T|\}$ **do**
- 25 **if** *minimal dissimilarity* **then**
- 26 $a_k(t) = \min((G_k^{0.05}(t))', (G_k^{0.50}(t))', (G_k^{0.95}(t))')$
- 27 **else if** *minimal squared dissimilarity* **then**
- 28 $a_k(t) = \min(((G_k^{0.05}(t))')^2, ((G_k^{0.50}(t))')^2, ((G_k^{0.95}(t))')^2)$
- 29 **else if** *minimal equi-dissimilarity* **then**
- 30 $a_k(t) = (G_k^{0.05}(t))' + (G_k^{0.50}(t))' + (G_k^{0.95}(t))'$

3.3 s-AWARE accuracy computation

The evaluation of crowdsourcing methods is based on comparisons between measures based on crowd ground truth and measures computed on a trusted ground truth (e.g. NIST assessors), also called Gold standard. Based on this known assumption, we can state that judgements given by an assessor will be correct if they are equal to gold judgements given by experts.

As discussed in chapter 2, trustworthy data can be effectively exploited to compute accuracy weights [5, 7].

In s-AWARE approaches we aim to combine the methodology of AWARE framework and the benefits of supervised weighting techniques, providing some supervised estimators for assessors' accuracies: the main idea behind s-AWARE approaches is that accuracy scores for assessors are computed with dissimilarities between each assessor's measures and gold measures (Figure 3.5). The smaller the dissimilarity, the better accuracy is assigned to the assessor.

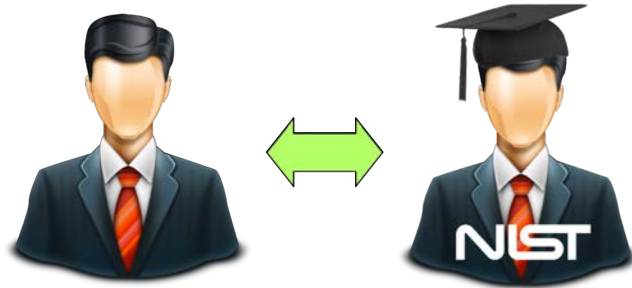


Figure 3.5: s-AWARE accuracy is computed with dissimilarity measures between crowd assessors and the gold standard on a training topicset

In order to execute s-AWARE approaches, then, a trusted dataset is needed to compute such scores. This prerequisite can be seen as a limit, because expert judgements are needed, even if for a small dataset. Nevertheless, as seen in 2.3, some test assessments are often given to crowd assessors in order to detect spammers, so data related to a small set of topics might be already computed, or can be computed with low effort: this will be called training topicset from now on.

Gold-crowd dissimilarities, called GAPs in analogy to dissimilarities computed

for u-AWARE approaches, are then normalized and directly used as accuracy scores. Algorithm 3 summarizes the shows the pseudo code to compute accuracy scores with s-AWARE.

3.3.1 GAPs and normalization

According to definition given in section 3.1, we describe how each measure is exploited to compute supervised GAPs.

GAPs are normalized in the $[0,1]$ range, but normalized GAP in s-AWARE has a different interpretation than normalized GAP in u-AWARE.

In u-AWARE, GAP is normalized to be a dissimilarity measure, that is an higher GAP means an higher dissimilarity with the random assessor and then an higher accuracy for the assessor.

In s-AWARE, on the contrary, an high GAP means high dissimilarity with the gold standard behaviour, then the normalization must map this GAP to a low accuracy value. Normalized GAP is used as accuracy score for the assessor, and can be interpreted as a measure of the closeness between crowd assessor and gold standard.

Let S be a set of IR systems and let T be the set of topics used in the training phase (subsequently training topicset): we define a $|T| \times |S|$ matrix containing the values of a performance measure computed on the runs generated by each system for each topic according to the judgements given by assessor k .

Frobenius Norm Frobenius norm GAP is computed as the norm of the difference between assessor's measures and gold measures

$$G_k = \| M_k - M^* \|_F$$

where M^* represent the gold measures.

High accuracy is obtained when norm assumes small values. To obtain values in the range $[0,1]$, the following normalization is applied:

$$G' = 1 - \frac{G}{\sqrt{|T| \cdot |S|}}$$

Root Mean Squared Error RMSE GAP is computed as the RMSE between assessor's measures and gold measures averaged by topic.

$$G_k = RMSE \left(\overline{M}_k(\cdot, S) - \overline{M}^*(\cdot, S) \right)$$

High accuracy is achieved when RMSE is low. To obtain values in the range [0,1], the following normalization is applied:

$$G' = 1 - G$$

Kullback-Leibler Divergence KLD GAP is computed as the KLD between the PDF of assessor measures and the PDF of gold measures.

$$G_k = D_{KL} (M_k(\cdot, \cdot) \| M^*(\cdot, \cdot))$$

KLD takes values in $[0, +\infty)$: in order to obtain values in the range [0,1], the following normalization is applied:

$$G' = e^{-\beta G}$$

where β is a positive real number.

Kendall Tau Correlation Tau GAP is computed as the correlation between the system rankings induced respectively by crowd and gold assessors' measures.

$$G_k = \tau (\overline{M}_k(\cdot, S), \overline{M}^*(\cdot, S))$$

To obtain values in the range [0,1], the following normalization is applied:

$$G' = |G|$$

AP correlation AP correlation GAP is computed as the correlation between the system rankings induced respectively by crowd and gold assessors' measures.

$$G_k = \tau_{AP} (\overline{M}_k(\cdot, S), \overline{M}^*(\cdot, S))$$

To obtain values in the range [0,1], the following normalization is applied:

$$G' = |G|$$

Algorithm 3: How to compute Assessor scores with s-AWARE

Data: T training topicset; $\hat{r}_t^k \forall t \in T$ ground truth generated by assessor k ; $\hat{r}_t \forall t \in T$ experts ground truth

Result: a_k accuracy score for assessor k

- 1 /* Compute the performance measures M_k for the k -th assessor and the gold standard measures M^* */
- 2 $M_k \leftarrow$ compute $m(\cdot)$ on \hat{r}_t^k ;
- 3 $M^* \leftarrow$ compute $m(\cdot)$ on \hat{r}_t ;
- 4 /* compute and normalize GAP G_k with respect to the gold standard */
- 5 **if** *frobenius norm* **then**
- 6 $G_k = \|M_k - M^*\|_F$ /* GAP computation;
- 7 $a_k = 1 - \frac{G}{\sqrt{|T| \cdot |S|}}$ /* [0,1] normalization;
- 8 **if** *RMSE* **then**
- 9 $G_k = RMSE(\overline{M}_k(\cdot, S) - \overline{M}^*(\cdot, S))$ /* GAP computation;
- 10 $a_k = 1 - G$ /* [0,1] normalization;
- 11 **if** *KL divergence* **then**
- 12 $G_k = D_{KL}(M_k(\cdot, \cdot) \| M^*(\cdot, \cdot))$ /* GAP computation;
- 13 $a_k = e^{-\beta G}$ /* [0,1] normalization;
- 14 **if** *Kendall Tau* **then**
- 15 $G_k = \tau(\overline{M}_k(\cdot, S), \overline{M}^*(\cdot, S))$ /* GAP computation;
- 16 $a_k = |G|$ /* [0,1] normalization;
- 17 **if** *AP correlation* **then**
- 18 $G_k = \tau_{AP}(\overline{M}_k(\cdot, S), \overline{M}^*(\cdot, S))$ /* GAP computation;
- 19 $a_k = |G|$ /* [0,1] normalization;

EXPERIMENTAL SETUP

In order to test the effectiveness of s-AWARE approaches, several experiments are performed. To get more accurate results, different experiments are performed considering each time a different set of topics as training set for s-AWARE, performing comparisons among approaches on the remaining part of the dataset, called test topicset or simply topicset from now on. Data from different topicsets is then averaged to compute the final analysis.

All the experiments are developed and run in MATLAB 2015a environment, exploiting MATTERS library ¹ for the common utilities useful for information retrieval tasks. In order to guarantee the reproducibility of the experiments, the source code of the experiments is available on BitBucket ².

This chapter is organized as follows: in section 4.1 we describe dataset, measures, approaches and parameters used for the experiments, in section 4.2 we present the experiments workflow followed to implement s-AWARE approach and to compare its effectiveness with a set of different other approaches.

¹<http://matters.dei.unipd.it/>

²<https://Lucapiaz@bitbucket.org/unipd-ferro-theses/piazzon.git>

4.1 Experimental parameters

The main purpose of our experiments is to evaluate the effectiveness of s-AWARE approach with respect to the following approaches:

- Majority vote (2.5.1)
- Expectation Maximization with MV seeding (2.5.2)
- AWARE with uniform accuracy weights
- sgl and tpc u-AWARE approaches with minimum squared dissimilarity weight(3.2)

First two approaches are the two different classic approaches usually taken as baseline, AWARE-uni is used to figure out if accuracy computation is effective, msd u-AWARE approaches are selected among all the u-AWARE approaches because, in most cases, msd weight performed better than other weighting techniques in the original paper [9].

The experiments are repeated several times, considering different combination of the following parameters:

- Kuple: is a set of assessors, whose data will be merged by each approach. The cardinality of a kuple is called kuple size.
- Evaluation Measure
- System/Run set: used interchangeably to indicate a set of ranked lists of documents, each of them generated by an IRS while searching relevant documents about a topic.
- Topicset: is the portion of the dataset (subset of the topics) used in every single experiment.

4.1.1 Dataset

4.1.1.1 Crowd assessors collection

As data for crowd assessors, we considered the data submitted to the TREC 21, 2012 Crowdsourcing track [52]. Research groups were asked to simulate the

relevance judgements given by the NIST assessors for 10 topics, selected from those of TREC 08. Topics IDs and descriptions are described in Table 4.1. A binary relevance judgement is given for each document in the judging pool of each topic.

	Topic ID	Description
1	411	Find information on shipwreck salvaging: the recovery or attempted recovery of treasure from sunken ships
2	416	What is the status of The Three Gorges Project?
3	417	Find ways of measuring creativity
4	420	How widespread is carbon monoxide poisoning on a global scale?
5	427	Find documents that discuss the damage ultraviolet (UV) light from the sun can do to eyes.
6	432	Do police departments use "profiling" to stop motorists?
7	438	What countries are experiencing an increase in tourism?
8	445	What other countries besides the United States are considering or have approved women as clergy persons?
9	446	Where are tourists likely to be subjected to acts of violence causing bodily harm or death?
10	447	What new developments and applications are there for the Stirling engine?

Table 4.1: Description for topics used in TREC 21 Crowdsourcing Track

The set of documents used in the experiments contains about 528K news documents from Financial Times (FT), Federal Register (FR), Foreign Broadcast Information Service (FBIS) and Los Angeles Times (LA) [52]. This dataset corresponds to disk 4 and 5 of the TIPSTER collection minus the Congressional Record.

In total 33 pools were submitted to TREC 21: we used 31 of them, excluding INFLB2012 and Orc2Stage because, for some topics, they did not assess any document as relevant. Pools information is reported in Table 4.3. Each pool, from now on is considered as a crowd assessor.

The gold standard of our experiments is the set of adjudicated relevance

	Topic ID	# Docs	NIST Rel	% REL	# Disag	% Disag	Rel to Non	Non to Rel	Total Rel
1	411	2056	27	1	15	1	1	1	27
2	416	1235	42	4	17	1	0	3	45
3	417	2992	75	3	60	2	3	3	75
4	420	1136	33	3	23	2	1	5	37
5	427	1528	50	3	42	3	14	1	37
6	432	2503	28	1	34	1	7	1	22
7	438	1798	173	11	118	7	16	5	162
8	445	1404	62	5	29	2	3	1	60
9	446	2020	162	9	119	6	15	9	156
10	447	1588	16	1	2	0	0	0	16

Table 4.2: Gold standard relevance judgements. From left to right: topic id, the total number of assessed documents by TREC participants, the number of NIST relevance judgements in the pool and its fraction, the number of NIST/assessors disagreements and its fraction, the number of NIST relevant documents finally labelled as non relevant, the number of NIST non relevant documents finally labelled as relevant, the total relevant documents per topic after the process

judgements of TREC 21, that are NIST judgements combined with the majority vote judgements from the submitted pools. In case of disagreement, documents have been manually assessed by TREC organizers. In table 4.2 is reported, for each topic, the fraction of relevant documents according to NIST and TREC organizers.

4.1.1.2 Retrieval Systems

Runsets utilized in our experiments came from the two TREC tracks which used the selected topics: the TREC 08 Ad-hoc track [53], which contains 129 runs and the TREC 13 Robust track [54], which contains 110 runs and whose goal was to test retrieval systems against hard topics.

4.1.2 Evaluation measures

Evaluation measures taken into consideration are a subset of the measures used in the original paper [9]:

- Average Precision (AP) (See Section 2.2.3.2) represents the most informative and stable measure in IR, since combines information from Precision and Recall in a top heavy way.

#	ID	Research Group
1	BUPTPRISZHS	Beijing University of Posts and Telecommunications
2	NEUEM1	Northeastern University
3	NEUElo2	Northeastern University
4	NEUElo3	Northeastern University
5	NEUElo4	Northeastern University
6	NEUElo5	Northeastern University
7	NEUNugget12	Northeastern University
8	Orc2G	University of Oxford and University of Southampton
9	Orc2GUL	University of Oxford and University of Southampton
10	Orc2GULConf	University of Oxford and University of Southampton
11	OrcVB1	University of Oxford and University of Southampton
12	OrcVB1Conf	University of Oxford and University of Southampton
13	OrcVBW16Conf	University of Oxford and University of Southampton
14	OrcVBW80	University of Oxford and University of Southampton
15	OrcVBW80Conf	University of Oxford and University of Southampton
16	OrcVBW9Conf	University of Oxford and University of Southampton
17	SSEC3excl	University of Oxford and University of Southampton
18	SSEC3incl	SetuServ
19	SSEC3inclML	SetuServ
20	SSECML2to99	SetuServ
21	SSECML50pct	SetuServ
22	SSECML75pct	SetuServ
23	SSML2pct	SetuServ
24	SSNoEC	SetuServ
25	UIowaS01r	University of Iowa
26	UIowaS02r	University of Iowa
27	UIowaS03r	University of Iowa
28	yorku12cs01	York University
29	yorku12cs02	York University
30	yorku12cs03	York University
31	yorku12cs04	York University

Table 4.3: List of the crowd Pools used un the experiments

- Normalized Discounted Cumulative Gain @ 20 (nDCG@20) (see section 2.2.3.3), meaning nDCG computed up to rank 20.

We used a log base $b=2$ and gains 0 and 5 respectively for non relevant and relevant documents.

4.1.3 Analysis measures

To compare the effectiveness of the different approaches we compared the evaluation measures computed with each approach against the measures computed on the gold standard. To do this we used two different methods:

- rank comparison: AP correlation (See section 3.1) is computed between

the ranking of the systems induced by the assessor, and the ranking of the system induced by gold standard. With AP correlation, we are interested in understanding to what extent each assessor can lead to the correct system ranking, for each evaluation measure.

- score comparison: RMSE (See section 3.1) is computed between the average assessor's measures and the average gold measures. With RMSE we want to figure out how accurate are each assessor's measures.

4.1.4 Parameters

4.1.4.1 Topicsets

In order to study how the dimension of the training set affects s-aware performance, we computed three group of test, with different topicset sizes:

- 7 training topics: we selected 35 different topicsets of 3 topics among the $\binom{10}{3}=120$ possible. For each topicset, the remaining 7 topics are used as training set for s-AWARE.
- 5 training topics: we selected 20 different topicsets of 5 topics among the $\binom{10}{5}=252$ possible. In this case, s-Aware and u-AWARE accuracy computation is performed on the same amount of data.
- 3 training topics: we selected 15 different topicsets of 7 topics among the $\binom{10}{7}=120$ possible.

The list of the topicsets is reported in Tables 4.4, 4.5 and 4.6.

4.1. EXPERIMENTAL PARAMETERS

Topicset	Topic IDs		
1	411	416	417
2	411	416	438
3	411	416	446
4	411	416	447
5	411	417	420
6	411	417	438
7	411	420	447
8	411	427	445
9	411	427	446
10	411	420	445
11	416	417	438
12	416	420	438
13	416	427	445
14	416	432	446
15	416	438	445
16	416	445	446
17	417	420	427
18	417	420	438
19	417	420	445
20	417	427	446
21	417	432	445
22	417	438	447
23	417	446	447
24	420	427	432
25	420	432	446
26	420	432	447
27	420	438	447
28	427	432	438
29	427	432	446
30	427	432	447
31	427	445	446
32	427	445	447
33	427	446	447
34	432	438	445
35	432	445	447

Table 4.4: topicsets used in experiments with 7 training topics and 3 test topics

Topicset	Topic IDs				
1	411	416	417	427	446
2	411	416	420	438	447
3	411	416	427	438	445
4	411	416	445	446	447
5	411	417	427	432	445
6	411	417	432	438	446
7	411	420	427	432	446
8	411	420	427	438	447
9	411	420	432	446	447
10	416	417	427	438	446
11	416	417	432	438	445
12	416	420	438	445	447
13	416	420	438	446	447
14	416	420	427	432	445
15	416	427	438	445	447
16	417	420	432	445	447
17	417	420	432	446	447
18	417	420	438	446	447
19	417	427	445	446	447
20	427	432	438	445	446

Table 4.5: topicsets used in experiments with 5 training topics and 5 test topics

Topicset	Topic IDs						
1	411	416	417	420	427	432	438
2	411	416	417	420	432	438	447
3	411	416	417	420	445	446	447
4	411	416	417	427	445	446	447
5	411	416	420	427	432	445	446
6	411	416	420	432	445	446	447
7	411	417	420	427	432	438	445
8	411	417	420	432	438	445	446
9	411	420	427	432	438	446	447
10	411	420	427	432	438	446	447
11	416	417	420	427	438	445	446
12	416	417	420	427	438	445	447
13	416	417	427	432	438	446	447
14	416	417	432	438	445	446	447
15	417	427	432	438	445	446	447

Table 4.6: topicsets used in experiments with 3 training topics and 7 test topics

4.1.4.2 Kuples

For both AP and nDCG, for each topicset we merged measures from different sets of assessors, of cardinality $k=2,3,\dots,30$. For each value of k , 100 kuples are randomly selected between the $\binom{31}{k} = \frac{31!}{k!(31-k)!}$ possible kuples.

4.1.4.3 Other parameters

In u-AWARE approaches, we considered 100 replicates for each of the three classes of random assessors (probability of relevance $p \in 0.05, 0.50, 0.95$).

For the computation of AP correlation in the case of ties, we sample and average over 100 randomly generated orderings.

For the estimation of probability density of measures with KDE (see Section 3.1), we use 100 equally spaced bins in the range $[0,1]$, a Gaussian kernel, and a bandwidth $b=0.015$.

For the EMMV algorithm we set a limit of 1000 iterations and a tolerance of 10^{-3} .

4.2 Experimental workflow

In this section we describe the main steps of the experiments performed in order to compare the different methods.

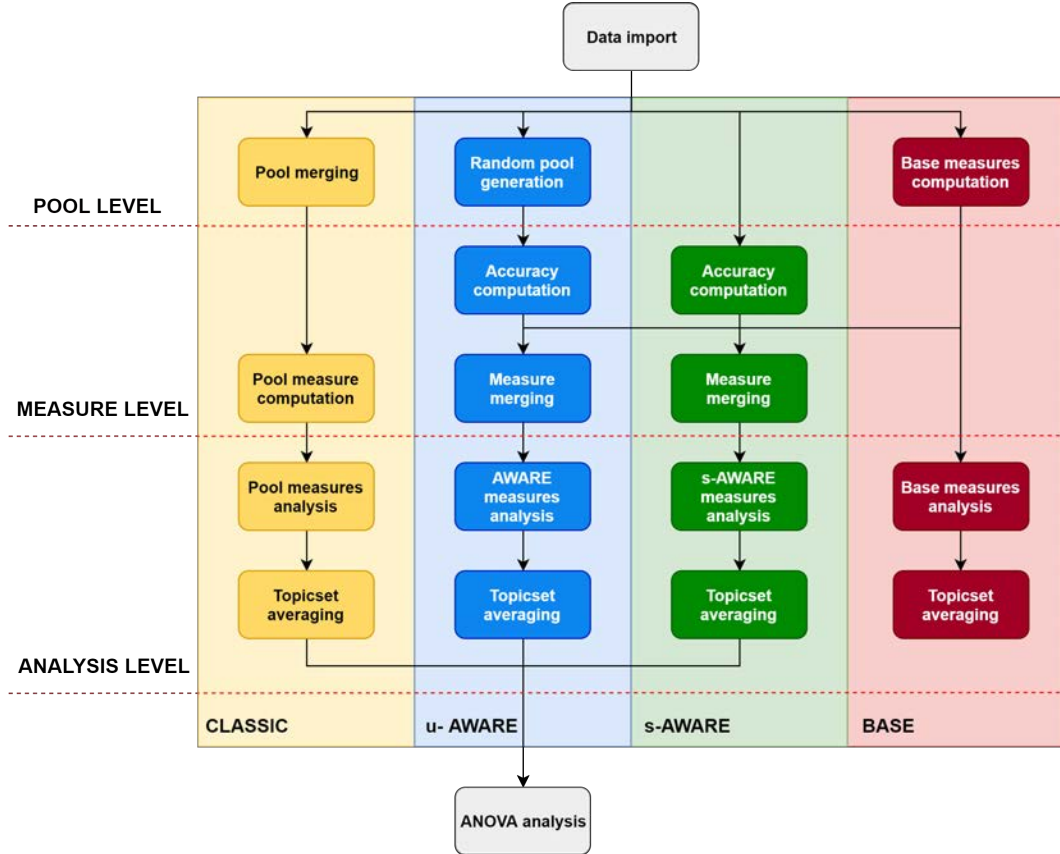


Figure 4.1: Experimental workflow

In Figure 4.1 is represented the workflow of the experiments that will be explained in the next sections:

- Data import: dataset is parsed and organized into data structures
- Base measures computation: Evaluation measures are computed with respect to crowd pools and gold pool. Crowd measures are compared against the gold measures to analyse the quality of the crowd assessors.
- Classic Approaches: in MV and EMMV approaches, a merged pool is performed. All the measures are first computed with respect to the ground

truth generated by the merged pool and then compared against the gold measures.

- **u-AWARE approaches:** in u-AWARE, measures are computed with respect to the ground truth generated by every single crowd assessor. Measures are then merged weighting with accuracy scores computed with Algorithms 1 and 2.
- **s-AWARE approaches:** in s-AWARE, measures are computed with respect to the ground truth generated by every single crowd assessor. Measures are then merged weighting with accuracy scores computed on training topicset with Algorithm 3.
- **ANOVA analysis:** a second layer of analysis, based on ANalysis Of VArance framework, is computed to summarize the results and to highlight strength and weaknesses of each approach.

4.2.1 Data import

The first step is the data import: Pool files and Run files are parsed and organized into data structures, Figure 4.2 and Algorithm 4 represent the import process.

For each topic, every assessor's pool is represented by a table containing document identifiers and relevance judgements given by the assessor.

For each set of systems, data of the runs is organized in a Topic-System table in which each cell contains the ranked list of document IDs performed by that system on that topic.

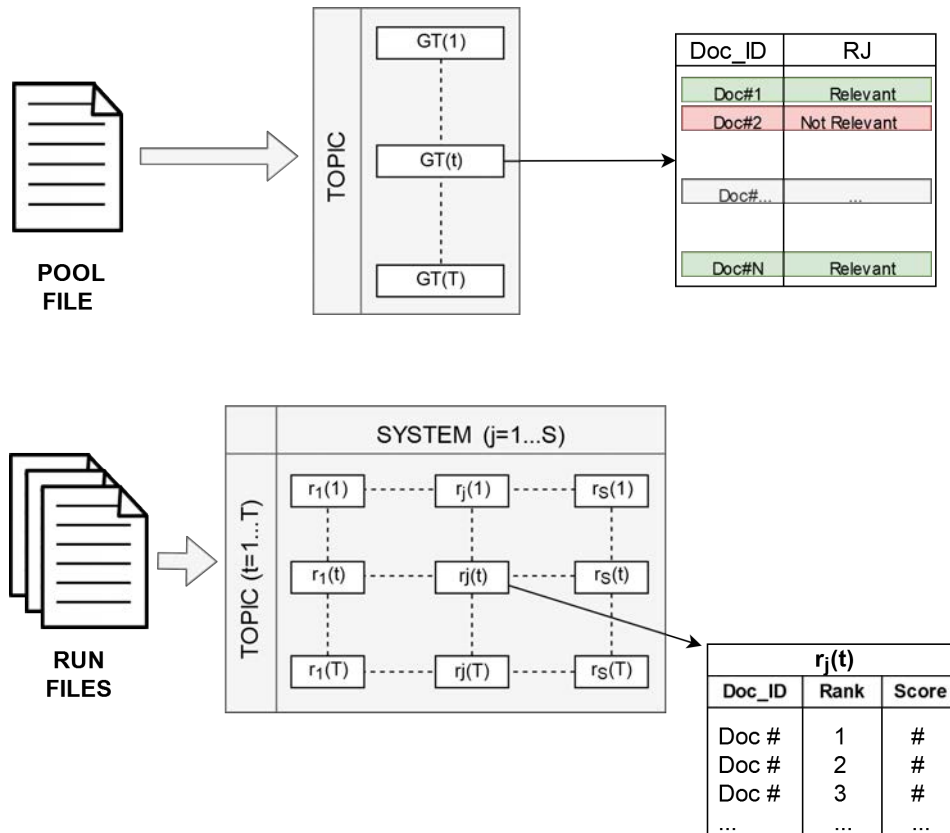


Figure 4.2: Data import: pools and runs are organized into data structures

Algorithm 4: Import collection pseudocode

```

1 /* Import Gold Pool*/;
2 goldpool  $\leftarrow$  structured version of GT
3 /* Import Crowd Pools*/;
4 P  $\leftarrow$  number of crowd pool files;
5 RS  $\leftarrow$  number of run sets;
6 for k  $\in \{1, \dots, P\}$  do
7   | poolk  $\leftarrow$  structured version of GTk
8   /* Import Runs*/;
9   for i  $\in \{1, \dots, RS\}$  do
10    | S  $\leftarrow$  number of run files in runset i;
11    for j  $\in \{1, \dots, S\}$  do
12      | runseti(j)  $\leftarrow$  structured version of run file

```

4.2.2 Base measures

Algorithm 5 describes the computation steps which involve the assessors' judgements: we first compute the evaluation measures on the runs to obtain the performance of the systems according to the gold standard ground truth and each crowd assessor's ground truth. This process is represented in figure 4.3.

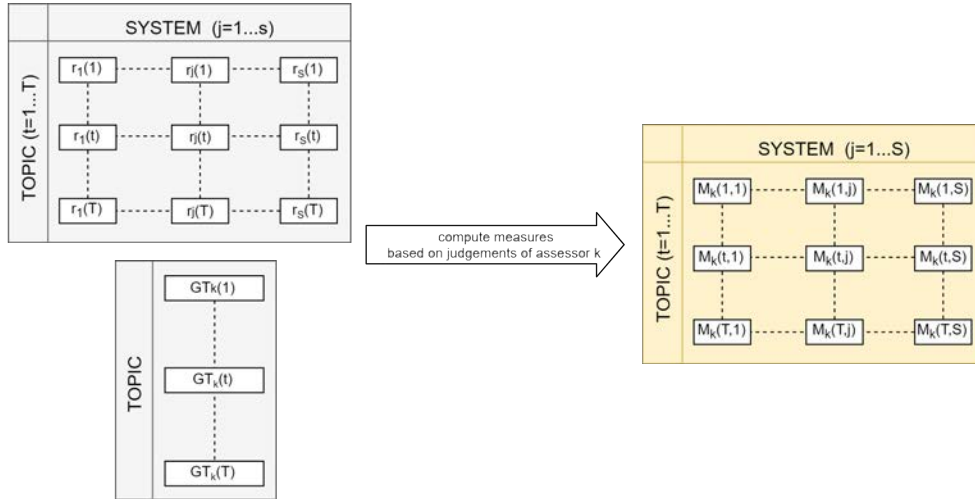


Figure 4.3: Base measure computation: for each assessor k , measures on runs are computed taking the assessor's judgements as gold standard

Base measure analysis Within each topicset, each assessor's measures are then averaged by topic and compared with the average gold measures: looking at figure 4.4, each column of the orange matrix is compared to the

gold measures column. Comparison is done with AP Correlation and RMSE, as discussed in the previous section.

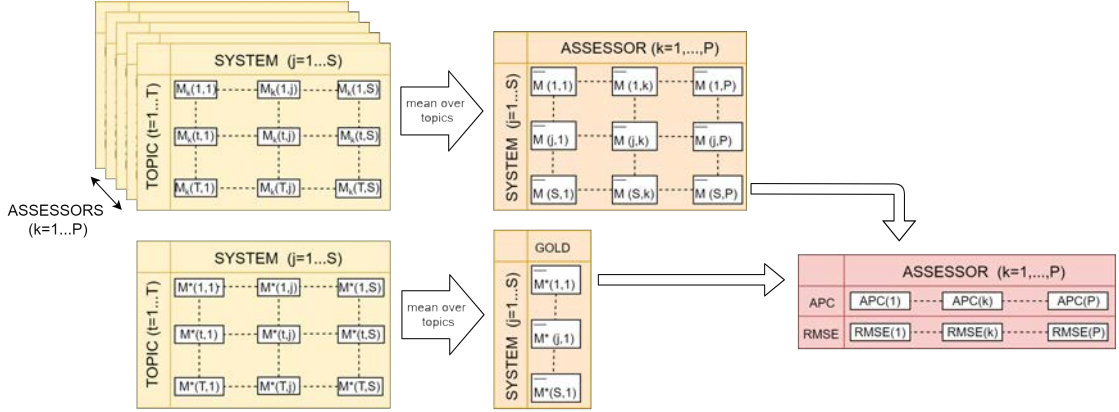


Figure 4.4: Base measure analysis: measures are averaged by topic and analysed with AP Correlation and RMSE

Average analysis over topicsets In order to summarize the analysis, an average over the topicsets is computed. In figure 4.5 is represented this process: each cell in the table on the right is computed as the arithmetic mean of the corresponding cells of topicset tables on the left.

In the next chapter, we will use this analysis on the input data to better understand the behaviour of each merging approach.

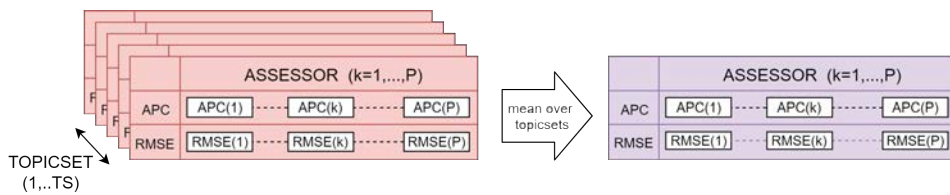


Figure 4.5: APC and RMSE average over the topicsets

Algorithm 5: Compute base measures pseudocode

```

1   $P \leftarrow$  number of crowd pools;
2   $RS \leftarrow$  number of run sets;
3  for  $i \in 1, \dots, RS$  do
4      foreach Measure do
5          /* Compute measures with respect to gold pool */;
6           $M^* \leftarrow \text{Measure}(\text{goldpool}, \text{runset}_i)$ 
7          /* Compute measures with respect to crowd pool */;
8          for  $k \in 1, \dots, P$  do
9               $M_k \leftarrow \text{Measure}(\text{pool}_k, \text{runset}_i)$ ;
10 /* Analyse base measures */;
11 for  $ts \in 1, \dots, TS$  do // for each topicset
12      $\overline{M}_{ts}^* \leftarrow \text{mean}(M^*, \text{topicset}_{ts})$  // average gold measures on current topicset;
13     for  $i \in 1, \dots, RS$  do // for each runset
14         foreach Measure do
15             for  $k \in 1, \dots, P$  do // for each assessor
16                 /*average base measures over topics in the current topicset */;
17                  $\overline{M}_{k,ts} \leftarrow \text{mean}(M_k, \text{topicset}_{ts})$ ;
18                 /*compute AP correlation and RMSE between base average measures and
19                     gold average measures*/;
19                  $APC_{ts}(k) \leftarrow \text{AP-correlation}(\overline{M}_{k,ts}, \overline{M}_{ts}^*)$ ;
20                  $RMSE_{ts}(k) \leftarrow \text{RMSE}(\overline{M}_{k,ts}, \overline{M}_{ts}^*)$ ;
21 /* Average analysis over topicsets */;
22 for  $i \in 1, \dots, RS$  do // for each runset
23     foreach Measure do
24         for  $k \in 1, \dots, P$  do // for each assessor
25              $APC(k) \leftarrow \text{mean}(APC_{ts}(k))$ ;
26              $RMSE(k) \leftarrow \text{mean}(RMSE_{ts}(k))$ ;

```

4.2.3 Classic Approaches

Both Majority vote and Expectation Maximization algorithms aim to create a merged ground truth at pool level. In order to compare s-AWARE with this methods, we compute measures based on MV and EMMV pools to be compared with measures obtained by s-AWARE approaches.

In the following paragraphs we explain the main computation steps, Algorithm 6 describes the entire process.

Pool merging For each kuple x , we consider the crowd pools relative to the assessors in the kuple. A merged pool is computed for each kuple, according to Majority Vote an Expectation Maximization methods, explained in Sections 2.5.1 and 2.5.2.

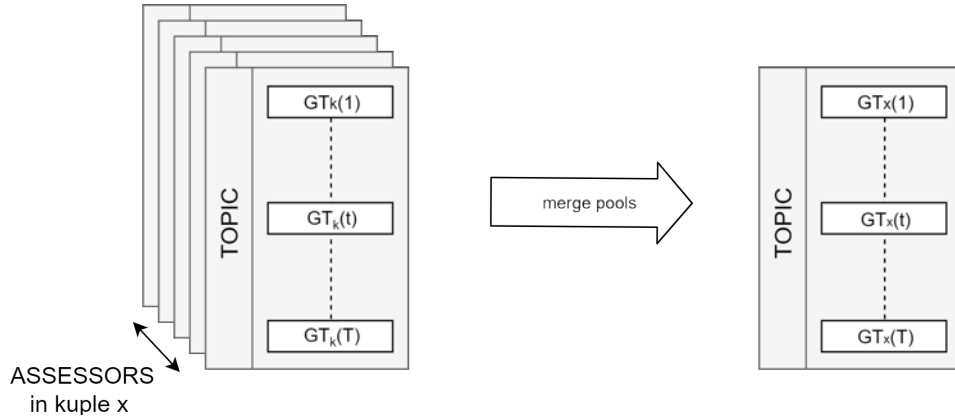


Figure 4.6: Pool merging: a single merged pool is computed using the relevance judgements coming from all the assessors

Pool measures computation For each kuple x , all the runs are evaluated based on the ground truth generated by the assessors in the kuple (Figure 4.7).

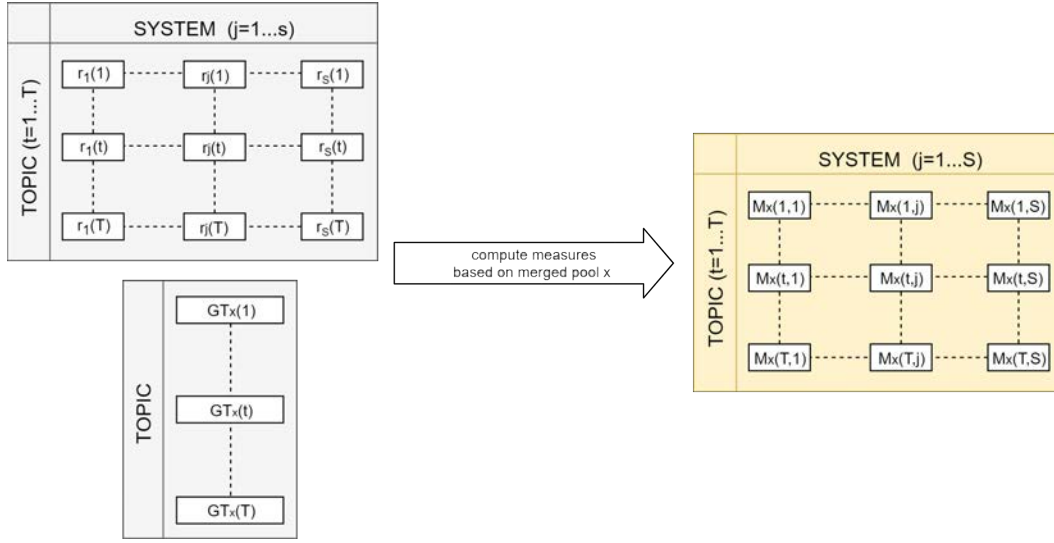


Figure 4.7: Pool measure computation: for each kuple, measures are computed taking the merged pool of the kuple as ground truth

Pool measures analysis Within each topicset, measures are averaged by topic and compared with the average gold measures (Figure 4.8). Comparison is done with AP correlation and RMSE to understand to what extent the crowdsourcing approach can lead to the same ranking of systems (APC) and similar scores (RMSE), for each evaluation measure.

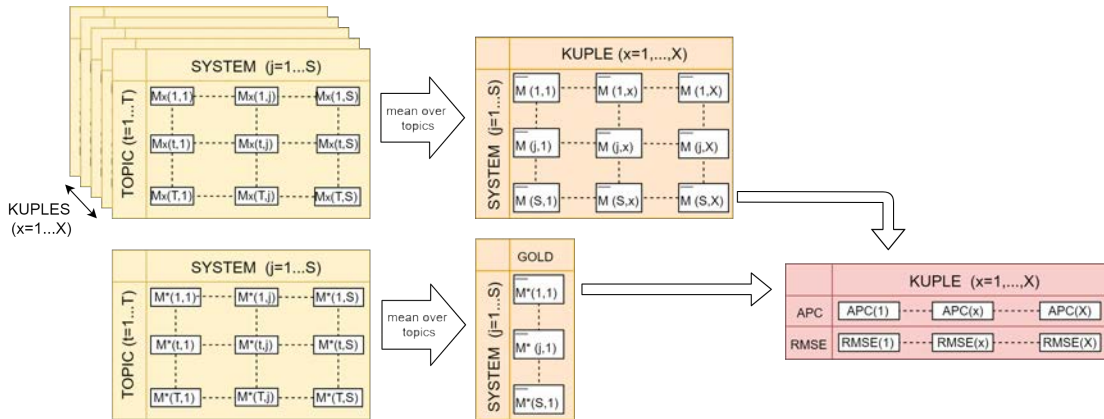


Figure 4.8: Pool measure analysis: measures are averaged by topic and analysed with AP Correlation and RMSE

Average analysis over topicsets In order to summarize the the measures analysis and run ANOVA analysis, an average over the topicsets is computed. In figure 4.9 is represented this process.

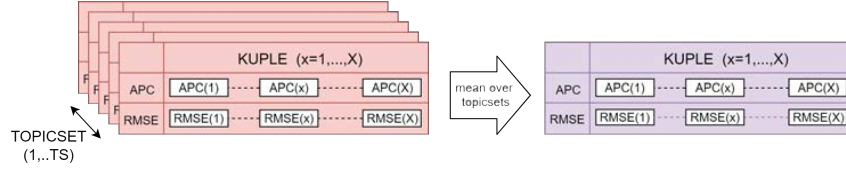


Figure 4.9: APC and RMSE average over topicsets

Algorithm 6: Classic approaches pipeline pseudocode

```

1   $P \leftarrow$  number of crowd pools;
2   $RS \leftarrow$  number of run sets;
3   $X \leftarrow$  number of kuples  $\forall$  kuple size ;
4   $TS \leftarrow$  number of topic sets;
5  /* Merge Pools */
6  foreach kuple size do
7      for  $x \in 1, \dots, X$  do // for each kuple
8           $GT_x \leftarrow$  MergePools({pools in kuple x})
9  /* Compute measures with respect to merged pools */
10 foreach kuple size do
11     for  $x \in 1, \dots, X$  do // for each kuple
12         for  $i \in 1, \dots, RS$  do // for each runset
13             foreach Measure do
14                  $M_x \leftarrow$  Measure( $GT_x, runset_i$ );
15 /* Analyse merged pools measures */;
16 for  $ts \in 1, \dots, TS$  do // for each topicset
17      $\bar{M}_{ts}^* \leftarrow$  mean( $M^*, topicset_{ts}$ ) // average gold measures on current topicset;
18     for  $i \in 1, \dots, RS$  do // for each runset
19         foreach Measure do
20             foreach kuple size do
21                 for  $x \in 1, \dots, X$  do // for each kuple
22                     /*average pool measures over topics in the current topicset */;
23                      $\bar{M}_{x,ts} \leftarrow$  mean( $M_x, topicset_{ts}$ );
24                     /*compute AP correlation and RMSE between pool average measures
25                     and gold average measures*/;
25                      $APC_{ts}(x) \leftarrow$  AP-correlation( $\bar{M}_{x,ts}, \bar{M}_{ts}^*$ );
26                      $RMSE_{ts}(x) \leftarrow$  RMSE( $\bar{M}_{x,ts}, \bar{M}_{ts}^*$ );
27 /* Average analysis over topicsets */;
28 for  $i \in 1, \dots, RS$  do // for each runset
29     foreach Measure do
30         foreach kuple size do
31             for  $x \in 1, \dots, X$  do // for each kuple
32                  $APC(x) \leftarrow$  mean( $APC_{ts}(x)$ );
33                  $RMSE(x) \leftarrow$  mean( $RMSE_{ts}(x)$ );

```

4.2.4 u-AWARE and s-AWARE approaches

u-AWARE and s-AWARE have very similar pipelines, since both of them aim to merge assessors judgements at measure level, and differ only in the accuracy computation step. In the following paragraphs we explain the main steps of AWARE pipeline, highlighting some crucial differences: Algorithms 7 and 8 describe, u-AWARE and s-AWARE pipelines.

Accuracy computation Accuracy computation is the most important phase of AWARE pipelines. In our experiments, dataset is split into two parts, called training topicset and test topicset (or simply topicset).

In u-AWARE, three sets of random assessors are generated, and GAP is computed between each assessor's measures and the measures computed on each random assessor replicate. To compute this GAPs, only test topics are used, in order to be compared with s-AWARE results.

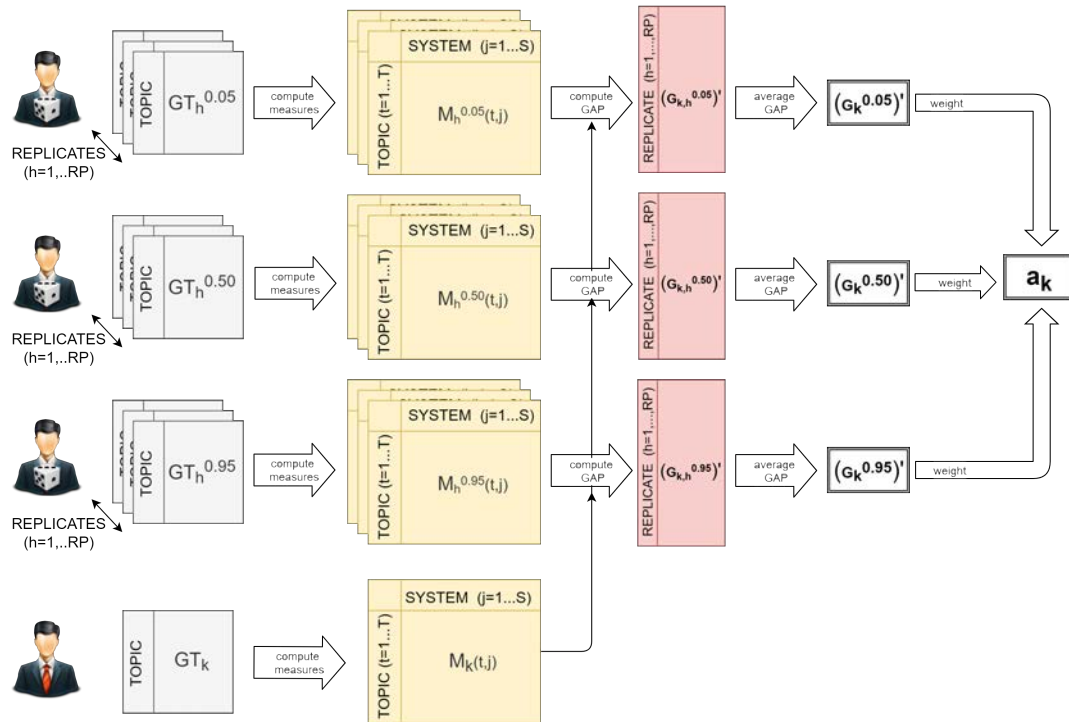


Figure 4.10: AWARE accuracy scores computation: GAPs are computed with respect to each random replicate. The mean GAPs are then combined with Weight computation

GAPs are then normalized and averaged by random replicate: this is done to guarantee a good estimate of random assessors. Accuracy computation is done combining the three random GAPs following algorithms 1 and 2: in sgl case we have an accuracy score for each assessor, in tpc case we have an accuracy score for each topic in the topicset.

In s-AWARE, GAP is computed between each assessor's measures and gold standard measures on training topics, following algorithm 3 described in Section 3.3.

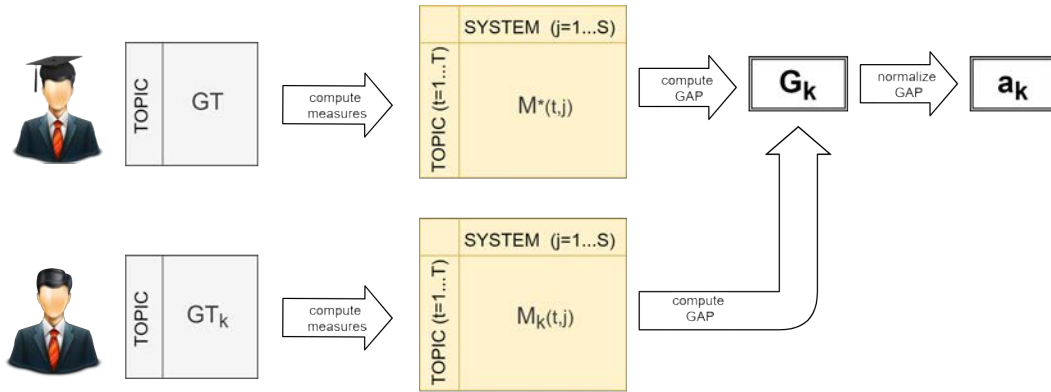


Figure 4.11: s-AWARE accuracy scores computation: normalized GAPs are directly used as accuracy scores

Measure merging For each kuple of assessors, measures are weighted with the so calculated accuracies and merged into a single matrix of measures. Measure merging is done on the test topicset.

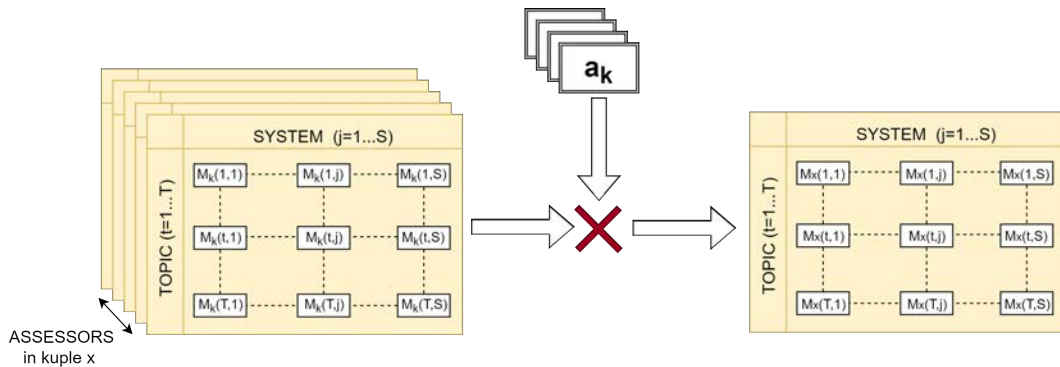


Figure 4.12: AWARE measure computation: merged measures are computed weighting assessors' by the accuracy scores

Pool measures analysis Kuple measures, similarly as previously described, are then averaged within the topicset and compared with gold measures. To do this, AP correlation and RMSE between the average measures is computed. It's important to highlight that even if this computation is similar to GAP computation, here the usage of RMSE and APC has a different meaning: in GAP computation, the goal is to get a dissimilarity measure to obtain each assessor accuracy, in analysis the goal is to understand how far the approach achieves similar results with respect to gold standard.

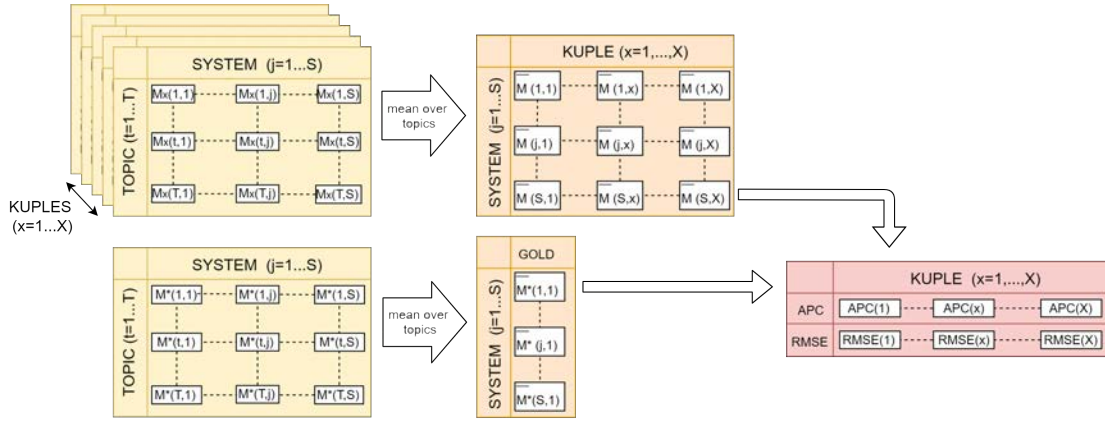


Figure 4.13: AWARE measures analysis: measures are averaged by topic and analysed with AP Correlation and RMSE

Average analysis over topicsets Analysis are then summarized averaging data over the topicsets. We obtain APC and RMSE mean values for each AWARE approach and s-AWARE approach, that will be analyzed with ANOVA.

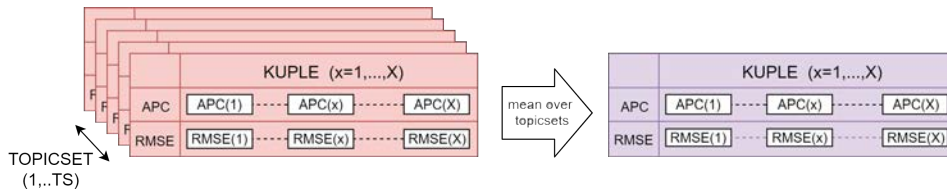


Figure 4.14: APC and RMSE average over topicsets

Algorithm 7: u-AWARE approaches pipeline pseudocode

```

1   $P \leftarrow$  number of crowd pools;
2   $RS \leftarrow$  number of run sets;
3   $X \leftarrow$  number of kuples  $\forall$  kuple size ;
4   $TS \leftarrow$  number of topic sets;
5   $RP \leftarrow$  number of random pool replicates for each class;
6  /* generate random Pools */
7  for ( $p \in \{0.05, 0.50, 0.95\}$ ) do
8      for ( $h \in \{1, \dots, RP\}$ ) do
9           $pool_h^p \leftarrow$  random judgements with probability of relevance= $p$ ;
10 for  $ts \in 1, \dots, TS$  do
11     for  $i \in \{1, \dots, RS\}$  do // for each runset
12         foreach Measure do
13              $\overline{M}_{ts}^* \leftarrow \text{mean}(M^*, \text{topicset}_{ts})$  // average gold measures in topicset;
14              $\text{weight} = \text{minimal squared dissimilarity}$ ;
15             for granularity  $\in \{sgl, tpc\}$  do
16                 for GAP  $\in \{fro, rmse, kld, tau, apc\}$  do
17                     /* Compute accuracy scores with algoritms 1 and 2*/
18                      $a_k \leftarrow \text{computeScores}(\text{granularity}, \text{GAP}, \text{weight}, ts)$ 
19                     /* Compute u-AWARE measures */
20                     foreach kuple size do
21                         for  $x \in 1, \dots, X$  do // for each kuple
22                             /* rescale accuracy scores */
23                              $\text{sum}_{ak} = \sum_{k \in x} a_k$ ;
24                             for  $k \in x$  do
25                                  $a'_k = \frac{a_k}{\text{sum}_{ak}}$ 
26                             /* compute u-aware measures */
27                              $M_x \leftarrow \sum_{k \in x} (M_k * a'_k)$ ;
28                             /*average u-AWARE measures in the current topicset */;
29                              $\overline{M}_{x,ts} \leftarrow \text{mean}(M_x, \text{topicset}_{ts})$ ;
30                             /*compute AP correlation and RMSE between aware average
31                                 measures and gold average measures*/;
31                              $APC_{ts}(x) \leftarrow \text{AP-correlation}(\overline{M}_{x,ts}, \overline{M}_{ts}^*)$ ;
32                              $RMSE_{ts}(x) \leftarrow \text{RMSE}(\overline{M}_{x,ts}, \overline{M}_{ts}^*)$ ;
33 /* Average analysis over topicsets */;
34 for  $i \in \{1, \dots, RS\}$  do // for each runset
35     foreach Measure do
36         for granularity  $\in \{sgl, tpc\}$  do
37             for GAP  $\in \{fro, rmse, kld, tau, apc\}$  do
38                 foreach kuple size do
39                     for  $x \in 1, \dots, X$  do // for each kuple
40                          $APC(x) \leftarrow \text{mean}(APC_{ts}(x))$ ;
41                          $RMSE(x) \leftarrow \text{mean}(RMSE_{ts}(x))$ ;

```

Algorithm 8: s-AWARE approaches pipeline pseudocode

```

1   $T \leftarrow$  all topics;
2   $P \leftarrow$  number of crowd pools;
3   $RS \leftarrow$  number of run sets;
4   $X \leftarrow$  number of kuples  $\forall$  kuple size ;
5   $TS \leftarrow$  number of topic sets;
6   $RP \leftarrow$  number of random pool replicates for each class;
7  for  $ts \in 1, \dots, TS$  do
8      for  $i \in \{1, \dots, RS\}$  do // for each runset
9          foreach Measure do
10              $\bar{M}_{ts}^* \leftarrow \text{mean}(M^*, \text{topicset}_{ts})$  // average gold measures in topicset;
11             for  $GAP \in \{fro, rmse, kld, tau, apc\}$  do
12                 /* Compute accuracy scores with algorithm 3*/
13                  $train_{ts} \leftarrow T - \text{topicset}_{ts}$ ;
14                  $a_k \leftarrow \text{computeScores}(GAP, train_{ts})$ ;
15                 /* Compute s-AWARE measures */
16                 foreach kuple size do
17                     for  $x \in 1, \dots, X$  do // for each kuple
18                         /* rescale accuracy scores */
19                          $sum_{ak} = \sum_{k \in x} a_k$ ;
20                         for  $k \in x$  do
21                              $a'_k = \frac{a_k}{sum_{ak}}$ 
22                         /* compute s-aware measures */
23                          $M_x \leftarrow \sum_{k \in x} (M_k * a'_k)$ ;
24                         /*average s-AWARE measures in the current topicset */;
25                          $\bar{M}_{x,ts} \leftarrow \text{mean}(M_x, \text{topicset}_{ts})$ ;
26                         /*compute AP correlation and RMSE between aware average
27                         measures and gold average measures*/;
27                          $APC_{ts}(x) \leftarrow \text{AP-correlation}(\bar{M}_{x,ts}, \bar{M}_{ts}^*)$ ;
28                          $RMSE_{ts}(x) \leftarrow \text{RMSE}(\bar{M}_{x,ts}, \bar{M}_{ts}^*)$ ;
29 /* Average analysis over topicsets */;
30 for  $i \in \{1, \dots, RS\}$  do // for each runset
31     foreach Measure do
32         for  $GAP \in \{fro, rmse, kld, tau, apc\}$  do
33             foreach kuple size do
34                 for  $x \in 1, \dots, X$  do // for each kuple
35                      $APC(x) \leftarrow \text{mean}(APC_{ts}(x))$ ;
36                      $RMSE(x) \leftarrow \text{mean}(RMSE_{ts}(x))$ ;

```

4.2.5 ANOVA analysis

In order to obtain more summary results and to evaluate how approaches performance is affected by the different combinations of kuple size, measure and systems, a second layer of analysis is computed, using ANalysis Of VAriance (ANOVA) methodology. With ANOVA we aim to investigate which are the main sources of variance within the analysis scores performed so far.

ANalysis Of VAriance ANOVA was introduced by Ronald Fisher in early 1900s [55] as a parametric statistical technique to be used to compare different distributions of data. This technique is based on the hypothesis testing done to understand whether or not the means of the different distributions are equal: the hypothesis testing model used in ANOVA analysis is so:

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \dots = \mu_n \\ H_1 : otherwise \end{cases}$$

where H_0 is called null hypothesis and H_1 is the alternative hypothesis. We reject null hypothesis if the different distributions are unlikely to be a realization for the null hypothesis (in this case, if means are "not equal") with respect to a chosen threshold α , called significance level, which represent the probability of wrongly label as statistically significant some means that are not. In order to compare distributions, F distribution is used [55].

In order to run ANOVA analysis, we have to compute the variable under examination for all the possible combinations of a set of parameters, also called factors. This data is then modelled with a GLMM (General Linear Mixed Model), where each value is seen as sum of two components, one is due to the Model and the other is an error.

$$Data = Model + Error$$

To explain how ANOVA works, we report a simple example of ANOVA [56] in which data depends only by a factor. Figure 4.15 represent the different values of the variable Y : each column of the table is dependent on a specific value of Factor A. There are n subjects for each distribution and p possible values for factor A.

		Factor A (Systems)				
		A_1	A_2	\cdots	A_p	
Subjects (Topics)	T'_1	Y_{11}	Y_{12}	\cdots	Y_{1p}	$\mu_{1\cdot}$
	T'_2	Y_{21}	Y_{22}	\cdots	Y_{2p}	$\mu_{2\cdot}$
	\vdots	\vdots	\vdots	Y_{ij}	\vdots	$\mu_{i\cdot}$
	T'_n	Y_{n1}	Y_{n2}	\cdots	Y_{np}	$\mu_{n\cdot}$
		$\mu_{\cdot 1}$	$\mu_{\cdot 2}$	$\mu_{\cdot j}$	$\mu_{\cdot p}$	$\mu_{\cdot\cdot}$

Figure 4.15: Different values of Y , relative to different combinations of topics and factor A values

The GLMM model, in this example is:

$$Y_{ij} = \mu_{\cdot\cdot} + \tau_i + \alpha_j + \epsilon_{ij}$$

where

- Y_{ij} is the single data point
- $\mu_{\cdot\cdot}$ is the grand mean of the data $\hat{\mu}_{\cdot\cdot} = \frac{1}{pn} \sum_{i=1}^n \sum_{j=1}^p Y_{ij}$
- τ_i is the effect on Y_{ij} given by i-th subject, estimated as $\hat{\tau}_i = \hat{\mu}_{i\cdot} - \hat{\mu}_{\cdot\cdot}$
- α_j is the effect of the j-th value of factor A, estimated as $\hat{\alpha}_j = \hat{\mu}_{\cdot j} - \hat{\mu}_{\cdot\cdot}$
- ϵ_{ij} is the error committed by the model in representing Y_{ij} , estimated as $\hat{\epsilon}_{ij} = Y_{ij} - (\hat{\mu}_{i\cdot} + \hat{\mu}_{\cdot j} - \hat{\mu}_{\cdot\cdot})$

If we had multiple samples for each point Y_{ij} , we could have separated from the error the interaction effects $\tau\alpha_{ij}$ between i-th subject and j-th factor, computed as

$$(\hat{\tau\alpha})_{ij} = \hat{\mu}_{ij\cdot} - (\hat{\mu}_{\cdot\cdot} + \hat{\tau}_i + \hat{\alpha}_j)$$

where $\hat{\mu}_{ij\cdot} = \frac{1}{|\text{replicates}|} \sum_{r=1}^{|\text{replicates}|} Y_{ijr}$

Starting from our model, we can rewrite it highlighting the different compo-

nents of the deviation from the grand mean:

$$\underbrace{Y_{ij} - \hat{\mu}_{..}}_{TotalEffects} = \underbrace{\hat{\mu}_{i.} - \hat{\mu}_{..}}_{SubjectEffects} + \underbrace{\hat{\mu}_{.j} - \hat{\mu}_{..}}_{FactorEffects} + \underbrace{Y_{ij} - (\hat{\mu}_{i.} + \hat{\mu}_{.j} - \hat{\mu}_{..})}_{ErrorEffects}$$

From this equation we can compute the Sum of Squares (SS) and the mean squares (MS) of different components as:

$$SS_{total} = SS_{subjects} + SS_{factor} + SS_{error} = \sum_{i=1}^n \sum_{j=1}^p (Y_{ij} - \hat{\mu}_{..})^2 \quad MS_{total} = \frac{SS_{total}}{DF_{total}}$$

$$SS_{subjects} = p \sum_{i=1}^n (\hat{\mu}_{i.} - \hat{\mu}_{..})^2 \quad MS_{subjects} = \frac{SS_{subjects}}{DF_{subjects}}$$

$$SS_{factor} = n \sum_{j=1}^p (\hat{\mu}_{.j} - \hat{\mu}_{..})^2 \quad MS_{factor} = \frac{SS_{factor}}{DF_{factor}}$$

$$SS_{error} = \sum_{i=1}^n \sum_{j=1}^p (Y_{ij} - (\hat{\mu}_{i.} + \hat{\mu}_{.j} - \hat{\mu}_{..}))^2 \quad MS_{error} = \frac{SS_{error}}{DF_{error}}$$

where DF indicates the degrees of freedom of the data, which are respectively $DF_{total} = pn - 1$, $DF_{subjects} = n - 1$, $DF_{factor} = p - 1$ and $DF_{error} = (p - 1)(n - 1)$. To find out whether the factor effect is statistically significant or not, we compute the F statistics defined as:

$$F_{factor} = \frac{MS_{factor}}{MS_{error}}$$

and we compare it with the F distribution with $(DF_{factor}, DF_{error})$ degrees of freedom. We search for the p value for which F_{factor} can be observed by chance under the null hypothesis, if this is lower than significance level α , means that null hypothesis is less probable than our minimum threshold and then we reject null hypothesis, saying that factor statistically influences the data.

If we had multiple factors under examination, for each factor we can compute a sum of squares and determine whether the factor is significant or not. For each combination of factors, we can then compute the interaction effects between factors, determining if the combination of two or more factors can lead to significant variance. This analysis is called k-way-ANOVA, where k is the number of considered factors.

Up to here, we just found out if a factor is statistically significant, that is if the different values for that factor cause a significant variation in means. We

can now compute a further step, investigating how much each factor affects the variance of the data. To do this we use Strength Of Association (SOA) coefficients.

We compute SOA as:

$$\hat{\omega}_{factor}^2 = \frac{DF_{factor}(F_{factor} - 1)}{DF_{factor}(F_{factor} - 1) + N}$$

where N is the number of points in our Grid of Points. This is an unbiased estimator of the influence of each factor: a common rule [55] states that $SOA \leq 0.06$ is considered a small effect, $0.06 \leq SOA \leq 0.14$ is considered a medium effect and $SOA \geq 0.14$ is considered a large effect.

Experiments' analysis In our experiments, we use AWARE analysis to determine how the different approaches behave with different combination of kuple sizes, evaluation measures and runsets. To summarize such analyses, we compute ANOVA analysis on APC and RMSE values.

For both APC and RMSE, we use then the following GLMM to compute three-way ANOVA with repeated measures analysis:

$$Y_{ijkl} = \underbrace{\mu_{....} + \kappa_i + \alpha_j + \beta_k + \gamma_l}_{MainEffects} + \underbrace{\alpha\beta_{jk} + \alpha\gamma_{jl} + \beta\gamma_{kl}}_{InteractionEffects} + \underbrace{\epsilon_{ijkl}}_{Error}$$

where:

- Y_{ijkl} is the single APC or RMSE value, given a combination of parameters
- $\mu_{....}$ is the grand mean
- κ_i is the kuple size ($x=2,...,30$)
- α_j is the effect of the j-th approach
- β_k is the effect of the k-th evaluation measure (AP, nDCG@20)
- γ_l is the effect of the l-th Runset (Trec 08, Trec 13)
- $\alpha\beta_{jk}$ in the interaction effect between approaches and measures
- $\alpha\gamma_{jl}$ in the interaction effect between approaches and runsets

- $\beta\gamma_{kl}$ in the interaction effect between measures and runsets
- ϵ_{ijkl} is the error committed by the model in predicting Y_{ijkl}

Figure 4.16 represents the ANOVA computation in our experiments, the average analysis for each approach, kuple size, measure and runset are analysed obtaining ANOVA analysis for APC and RMSE.

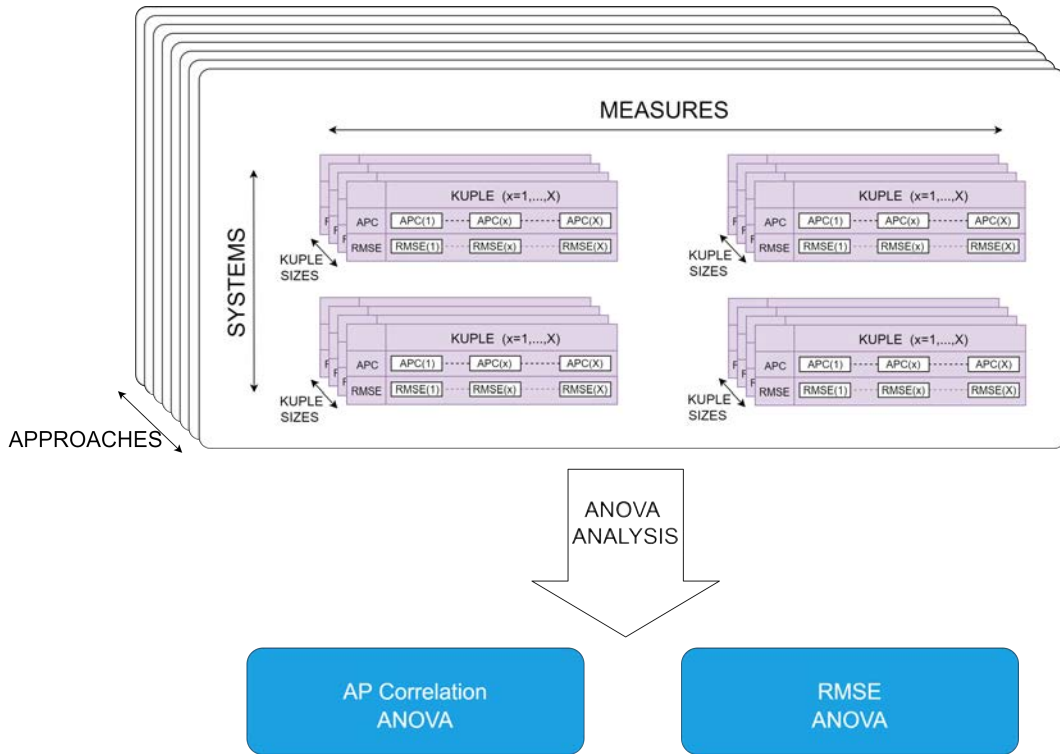


Figure 4.16: ANOVA analysis

EXPERIMENTAL RESULTS

In this chapter we describe the results coming from the experiments we described so far. We will highlight in which case s-AWARE methods outperform unsupervised methods and classic methods.

In section 5.1, we first look at the analyses on assessors' pools, which will be compared to our results. In section 5.2 we look at the analyses relative to the tests performed using an equal size for training and test set. In section 5.3, we analyse if different sizes of the training set affect the performance of supervised approaches.

5.1 Base measures analysis

5.1.1 AP Correlation

Figure 5.1 represents the mean AP correlation between each assessor's measures and gold measures, averaged across measures and runsets. This plot highlights that assessors have very different behaviours (we remind that what we call assessor, is it's actually a pool submitted to TREC21 Crowdsourcing track). Skierarchy pools, in green, perform better than all the others as expected, since they are obtained from a complex interaction process between crowd assessors, machine learning and experts (section 2.5.5). The worst pool in terms of system ranking is NEUEM1, obtained with Expectation Maximiza-

CHAPTER 5. EXPERIMENTAL RESULTS

tion algorithm as described in section 2.5.2.

Looking at figures 5.2 and 5.3 we observe that the most part of assessors ranks better the systems based on AP measures than nDCG measures, and that systems from TREC08 are ranked more efficiently by most of the assessors.

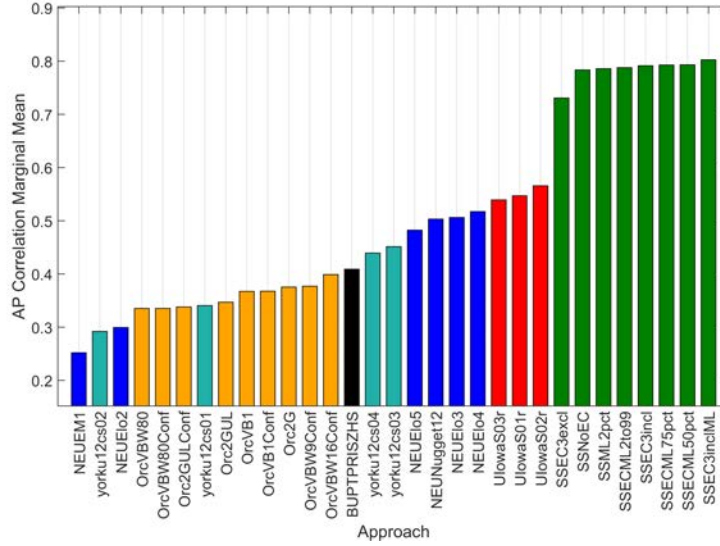


Figure 5.1: APC for base measures

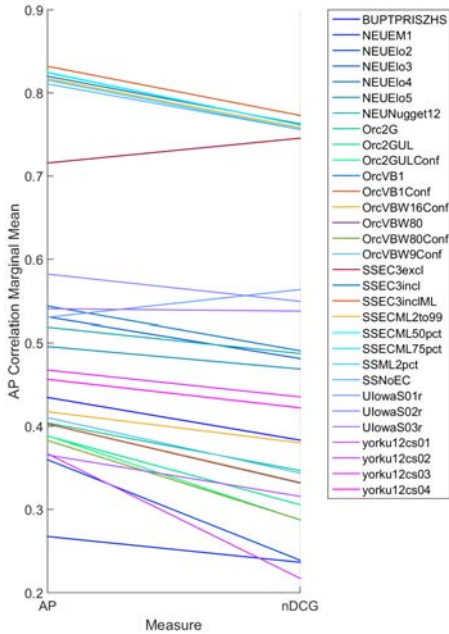


Figure 5.2: APC Assessor*Measure interaction for base measures

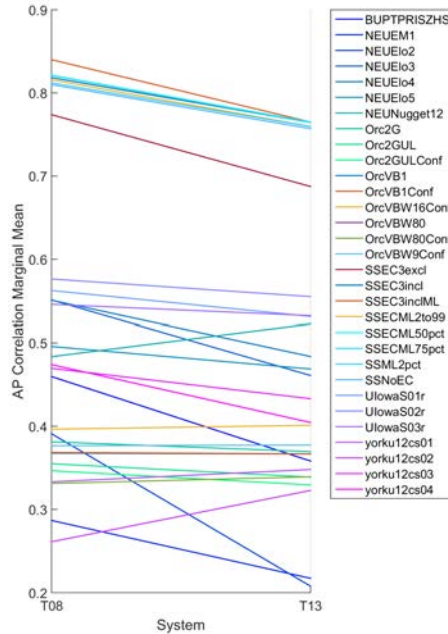


Figure 5.3: APC Assessor*Runset interaction for base measures

5.1.2 RMSE

When we look at the RMSE computed between assessor's measures and gold measures (Figure 5.4), we see that Skierarchy pools achieve the best performances and that Nothern Eastern University pools generally achieve bad results in terms of RMSE, with the exception of NEUNugget12 pool. NEU pools result to be bad in terms of RMSE but perform quite well in terms of APC: this probably means that NEU pools have the ability to correctly rank the top systems (AP correlation is top heavy), but are not good in estimating the real values of the gold measures.

Figure 5.5 shows that measure is affecting the assessor measure accuracy. For example, taking OrcVBW16Conf and NEUElo pools we can observe that OrcVBW16Conf leads to better results on AP measures than NEUElo pools, while this ones outperform OrcVBW16Conf in nDCG measures.

This behaviour strengthen the motivations behind AWARE: errors in pool computation affect in a different way the different evaluation measures. Merging assessors at measure level, we can take into account this phenomenon, otherwise neglected.

Figure 5.6 shows that the assessors measures computed on TREC08 runs are slightly more accurate than measures on TREC13.

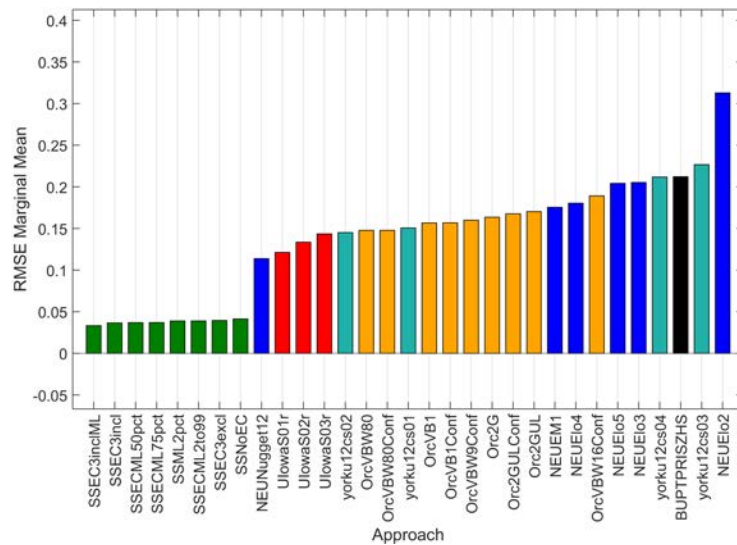


Figure 5.4: RMSE for base measures

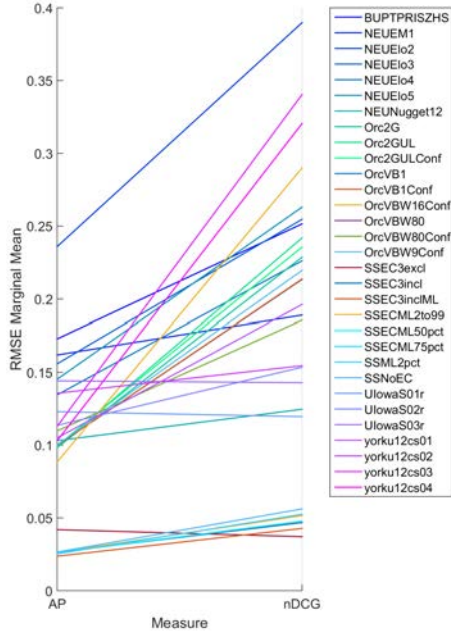


Figure 5.5: RMSE Assessor*Measure interaction for base measures

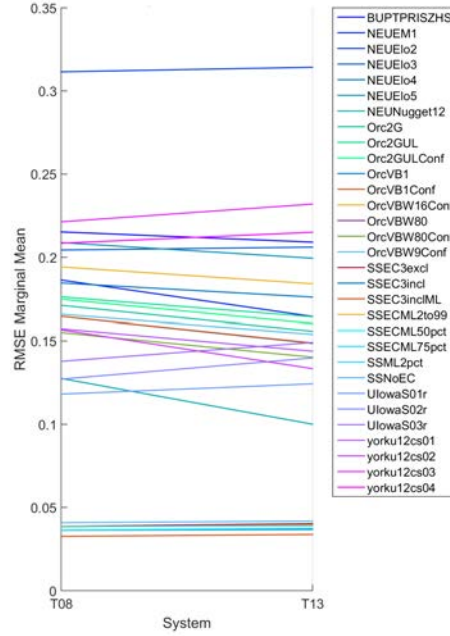


Figure 5.6: RMSE Assessor*Runset interaction for base measures

5.2 Results with equal Train-Test size

5.2.1 AP Correlation

Table 5.1 is the ANOVA table for AP Correlation: looking at SOA coefficients we see that all the considered factors are large size effects. The largest effect is the Approach effect, highlighting that different approaches can lead to very different performance. Measure and Systems effects are smaller than Approach effect, but their values support the intuition behind the AWARE framework of computing the merging of multiple assessors at measure level, when different measures and systems are taken into account.

This can be even better observed analysing the interaction effects: Approach*Measure effect is a large interaction effect, stating that different measures have a great and potentially different impact on each approach performance.

5.2. RESULTS WITH EQUAL TRAIN-TEST SIZE

	SS	DF	MS	F	p- value	SOA
K-uple Size	1,37679	28	0,04917	232,49984	<0.0001	
Approach	1,33129	17	0,07831	370,28459	<0.0001	0,75041
Measure	0,41956	1	0,41956	1983,84163	<0.0001	0,48708
Systems	0,37419	1	0,37419	1769,29570	<0.0001	0,45855
Approach*Measure	0,34205	17	0,02012	95,13682	<0.0001	0,43389
Approach*Systems	0,13625	17	0,00801	37,89693	<0.0001	0,23101
Measure*Systems	0,59299	1	0,59299	2803,86943	<0.0001	0,57308
Error	0,42404	2005	0,00021			
Total	4,99716	2087				

Table 5.1: ANOVA table for AP Correlation (5 test topics)

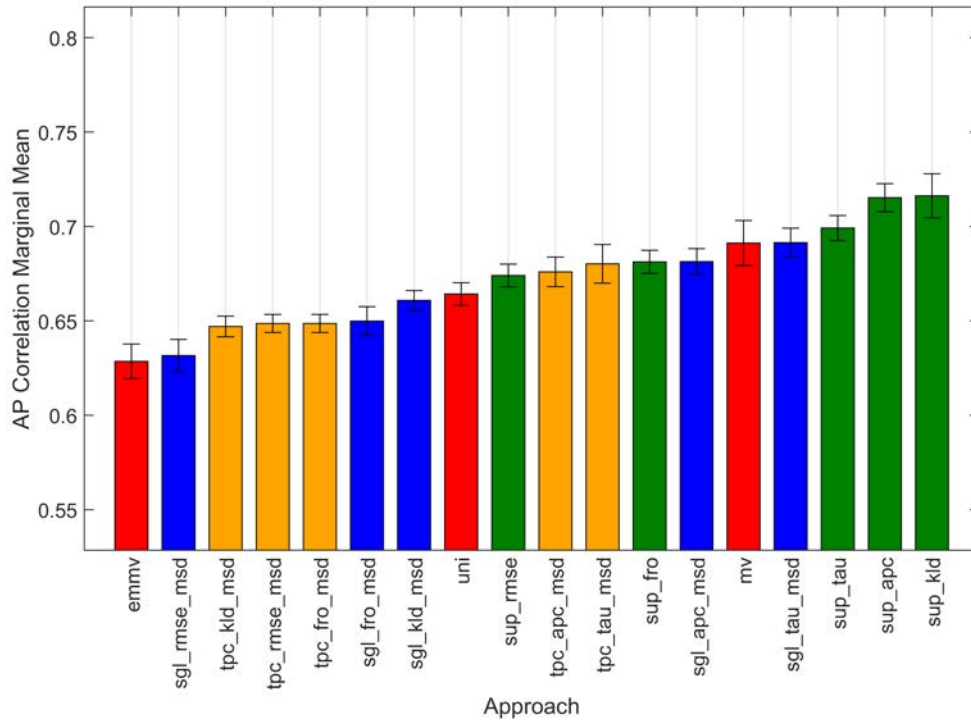


Figure 5.7: APC Approach main effect (5 test topics)

In Figures 5.7-5.10 are represented the marginal means of AP correlation values: each plot represents how the performance is affected by different approaches, different number of merged assessors, different measures and runsets.

Figure 5.7 show the average performance of each approach in terms of system ranking. Yellow and blue bars represent u-Aware approaches, green bars are s-Aware approaches and red bars are the baselines. We can see that all the s-AWARE methods outperform AWARE-uni and most of them perform better than u-AWARE approaches, meaning that the s-AWARE accuracy computation is effective. s-AWARE methods based on ranking and distribution GAPs perform better, in average, than all the other approaches.

In Figure 5.8 we can see that increasing the number of merged assessor improves the average performance. Figures 5.9 and 5.10 show that measures computed by the approaches follow the behaviour of the assessors measures.

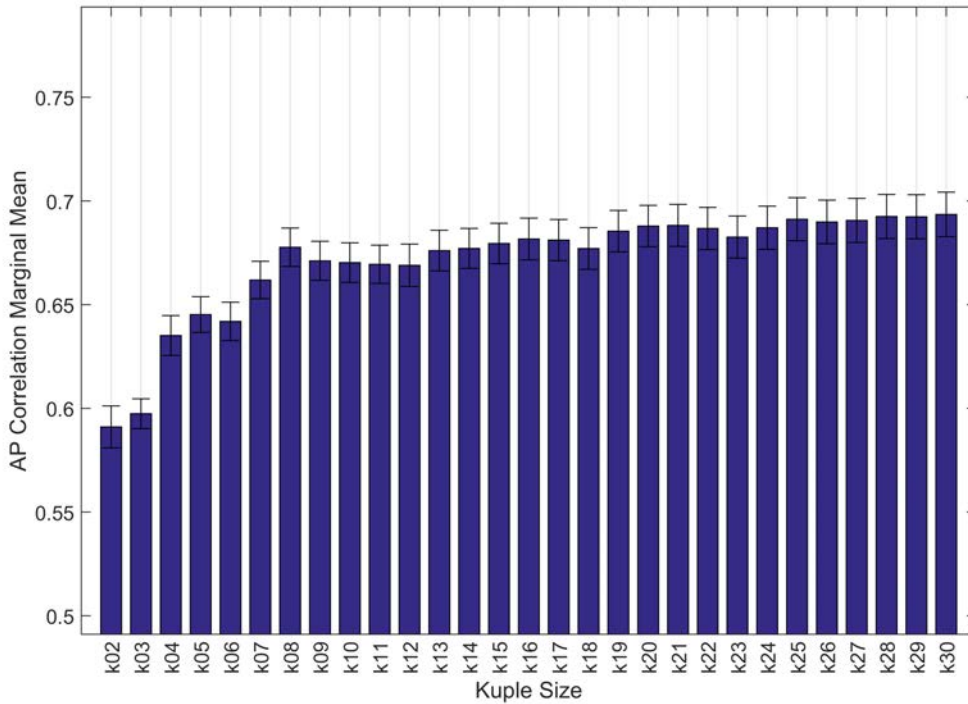


Figure 5.8: APC Kuple main effect (5 test topics)

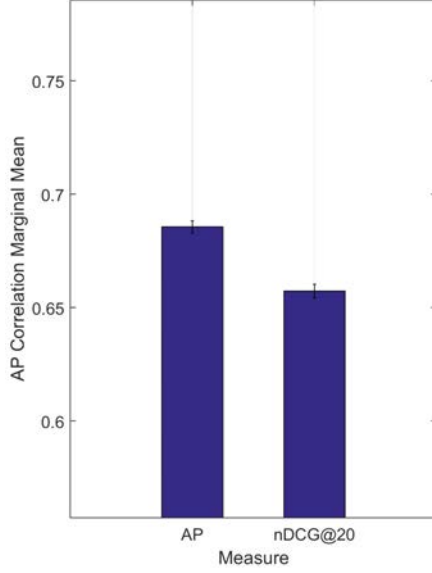


Figure 5.9: APC Measure main effect (5 test topics)

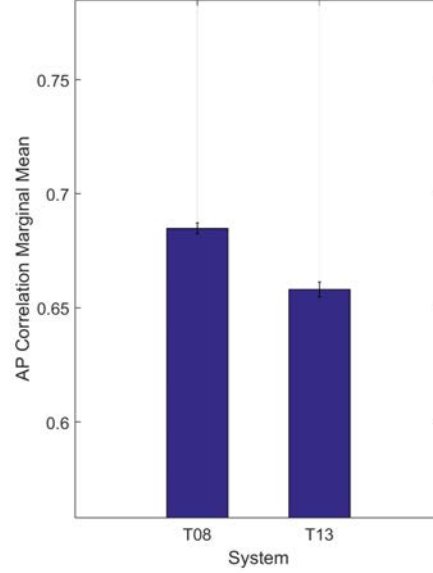


Figure 5.10: APC Runset main effect (5 test topics)

Figures 5.11-5.13 show how Kupple size, measures and runsets affect each approach performance in terms of AP Correlation: dashed lines are for single-score u-AWARE, dotted lines are for topic-by-topic u-AWARE, solid lines are for s-AWARE and thicker lines are for the baselines.

All the approaches benefit of the increasing number of merged assessors, as shown in figure 5.11. All s-AWARE approaches perform better than the uniform case even merging a small number of assessors. Sup_kld and sup_apc are the best approaches in terms of AP correlation, and all s-AWARE approaches generally outperform the corresponding u-AWARE approaches.

Majority vote reaches AWARE-uni results with kuples of 7 assessors, and perform better than all the other approaches when merging more than 24 assessors. This behaviour can be partially due to the nature of the gold standard: as described in section 4.1.1, the gold pool is created from NIST assessments and MV assessments, manually adjudicating the documents for which the majority vote of the pools and the NIST assessments disagreed. The performance of MV, in our experiments can then overestimate the real performances of MV. Expectation Maximization algorithm achieves the worst results, improving its performances only with a large number of merged assessors.

Both supervised and unsupervised approaches follow the same ascending trend,

achieving good behaviour even with small kuple sizes and ensuring a more stable behaviour.

Figure 5.12 shows the Approach*Measure interaction effect. Systems are ranked better if we look at AP measures than nDCG measures. In particular we note that sup_kld, sup_tau and and sup_apc perform better than MV or at least as MV for both measures; sup_kld, however, is the approach for which we have the largest difference between AP and nDCG performances. In Figure 5.13 we see that most approaches perform better on T08 runs than T13 runs.

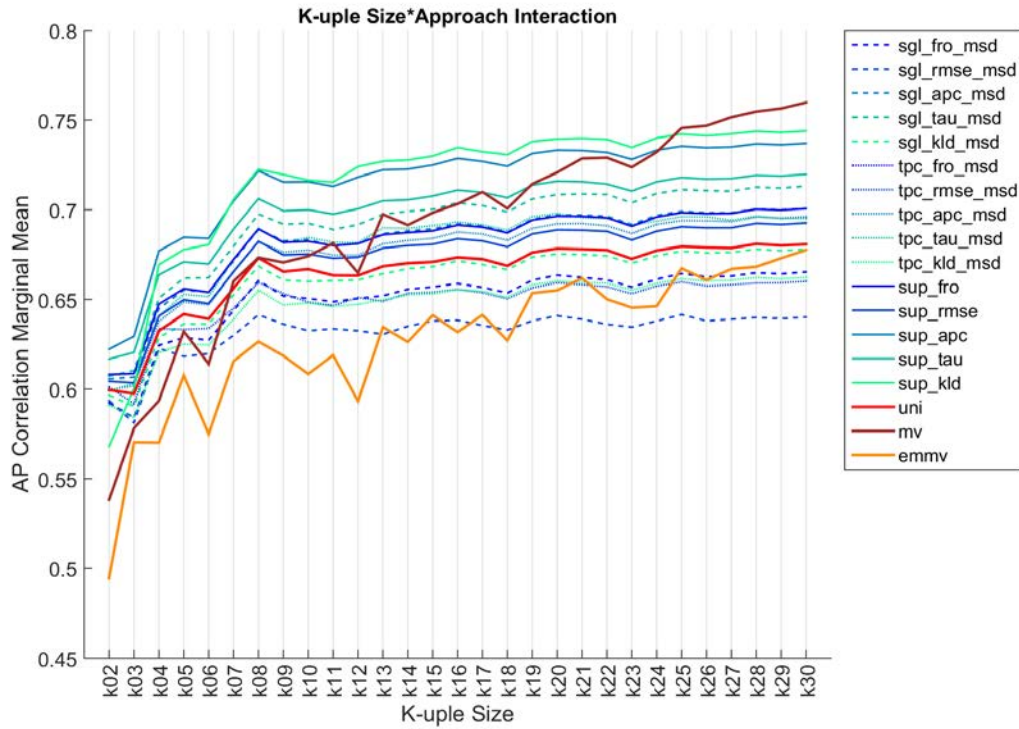


Figure 5.11: APC Approach*Kuple interaction effect (5 test topics)

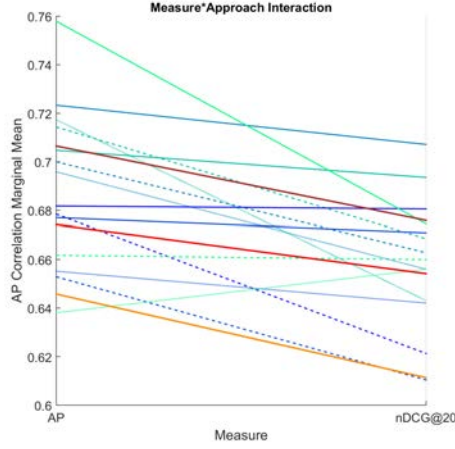


Figure 5.12: APC Approach*Measure interaction effect (5 test topics)

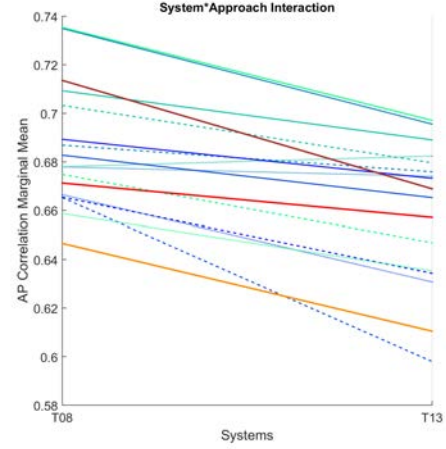


Figure 5.13: APC Approach*Runset interaction effect (5 test topics)

5.2.2 RMSE

Table 5.2 shows the ANOVA table for RMSE. Measure and Approach are the two largest main effects and Approach*Measure is the largest interaction effect. These three values confirm that approaches behave differently with different measures and then that AWARE methodology is hopeful.

Looking at the Approach main effects plot (Figure 5.14), we see that sup_apc and sup_tau confirm their good performance with respect to all the other approaches. All s-AWARE approaches perform better than the uniform case and most of them outperform Majority vote. In particular we can notice that sup_fro and sup_rmse behaves slightly better than MV, while perform worse in terms of AP correlation. A more evident behaviour is about sup_kld: when we look at RMSE, it performs as AWARE-uni and MV even if it was the best approach in terms of APC. This can be explained saying that sup_kld accuracy scores lead to a good system ranking for the top positions, but in general are not so effective in computing the real values for the measures.

In Figure 5.15 we can see that RMSE improves with the increasing number of merged assessors. Figures 5.16 and 5.17 show that measures computed by the approaches follow the behaviour of the assessors measures.

CHAPTER 5. EXPERIMENTAL RESULTS

	SS	DF	MS	F	p- value	SOA
K-uple Size	0,07822	28	0,00279	41,24387	<0.0001	
Approach	0,47972	17	0,02822	416,63970	<0.0001	0,77190
Measure	0,65403	1	0,65403	9656,47963	<0.0001	0,82220
Systems	0,03584	1	0,03584	529,10047	<0.0001	0,20187
Approach*Measure	0,55030	17	0,03237	477,94009	<0.0001	0,79521
Approach*Systems	0,01574	17	0,00093	13,67330	<0.0001	0,09353
Measure*Systems	0,00443	1	0,00443	65,47980	<0.0001	0,02996
Error	0,13580	2005	0,00007			
Total	1,95407	2087				

Table 5.2: ANOVA table for RMSE (5 test topics)

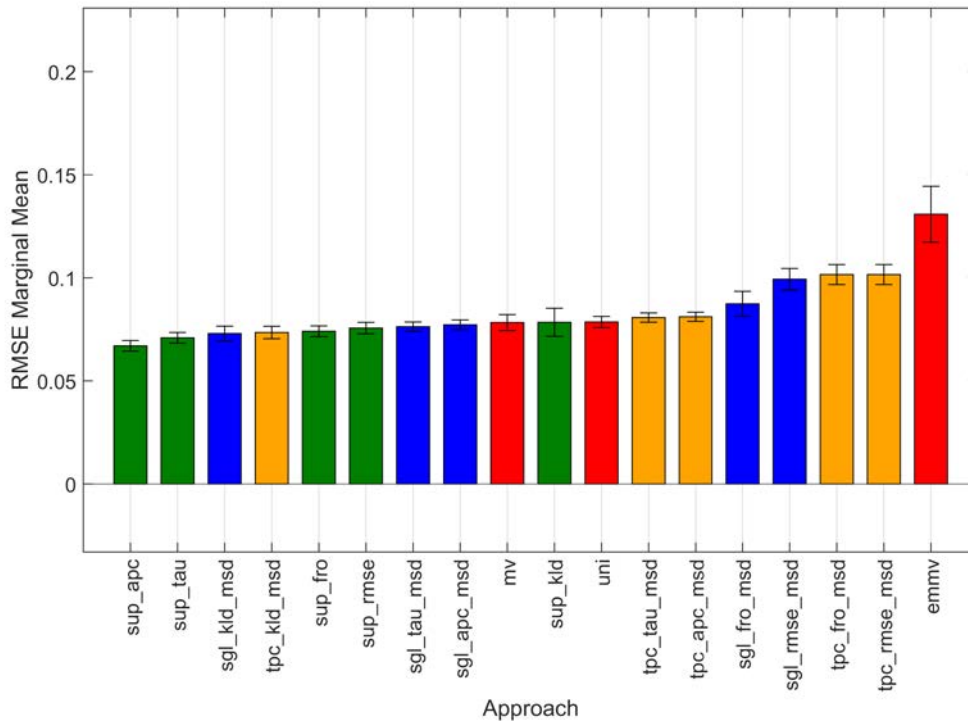


Figure 5.14: RMSE Approach main effect (5 test topics)

5.2. RESULTS WITH EQUAL TRAIN-TEST SIZE

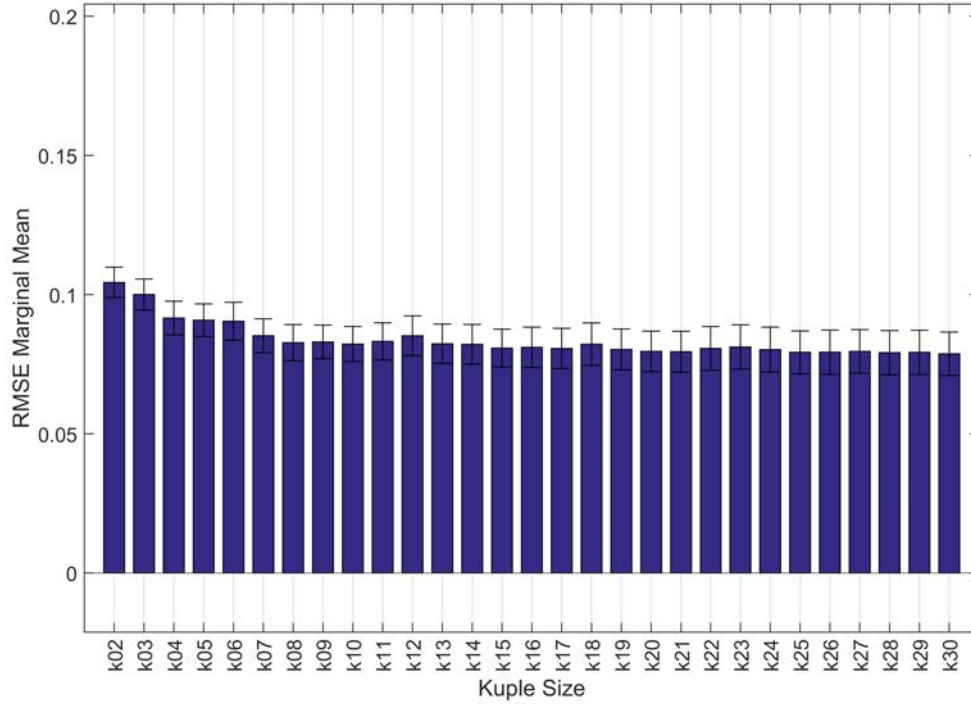


Figure 5.15: RMSE Kupple main effect (5 test topics)

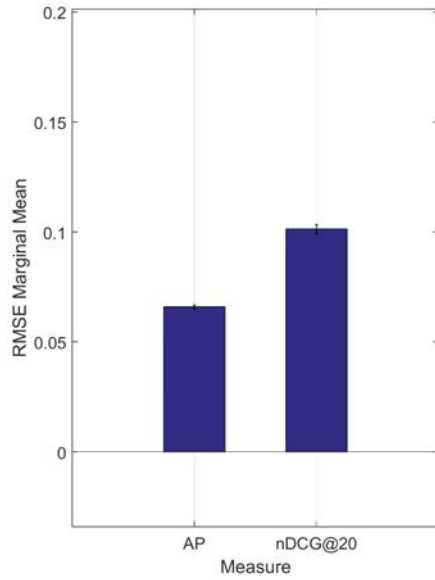


Figure 5.16: RMSE Measure main effect (5 test topics)

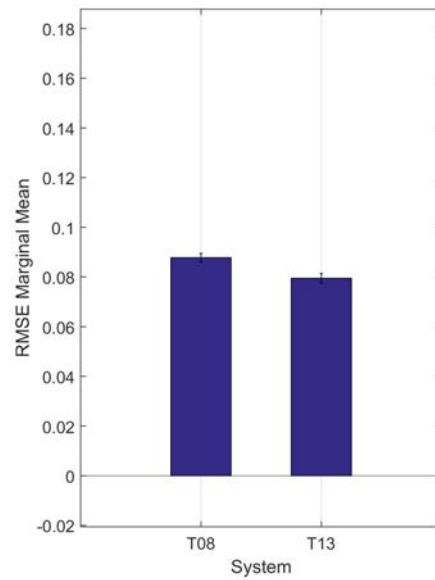


Figure 5.17: RMSE Runset main effect (5 test topics)

Looking at interaction effects (Figures 5.18-5.20) we can see that the majority of the approaches improve their performance increasing the kuple size, with the exception of Expectation maximization and u-AWARE approaches based on fro and rmse GAP. s-AWARE approaches, with the exception of sup_kld, perform better than AWARE-uni, MV and its unsupervised version. This confirms the intuition behind s-AWARE approaches: AWARE can outperform baseline approaches merging assessors at measure level, and in order to improve performance is useful to use information from a small set of trusted judgements instead of basing only to the non-random behaviour of assessors.

Sup_tau and sup_apc perform better than all the other approaches even with small kuples.

Figure 5.19 shows the Approach*Measure interaction effect. All the approaches achieve very similar performance with AP measures, but some of them (EM, sup_kld, sgl_fro, sgl_rmse and tpc_rmse) worsen their performance with nDCG measures. In particular sup_kld seems to be the best approach with AP measures, but underperform all the other s-AWARE approaches and most of the other approaches with nDCG measures.

In Figure 5.20 we see that most of the approaches achieve smaller values of RMSE for T13 runs, but this is probably due to the smaller value of the evaluation measures for T13 systems. Reasoning by comparison, we can say that most of the Approaches perform worse than Majority vote with T08 runs, and better than MV with T13 runs. Sup_tau and sup_apc perform better than all the other approaches with both T08 and T13 runs.

Summarizing the results explained so far, we can say that:

- s-AWARE approaches behave better than the uniform case, proving the effectiveness of supervised accuracy computation
- s-AWARE approaches usually perform better than the corresponding u-AWARE approaches using the same GAP measure
- s-AWARE approaches outperform Majority vote while estimating the real value of the measures, while MV achieves a better performance in terms of the system ranking when we merge a large number of assessors

- the best s-AWARE approaches are sup_kld, sup_tau and sup_apc: sup_kld achieves different performance with different combination of the factors and is better in determining the top systems than in estimating the real value for the measures. Sup_apc seems to be the most stable measure, achieving always the best results when looking at RMSE and very good results if we consider AP correlation analysis. Sup_tau follows sup_apc behaviour, achieving slightly worse performance.

This good results in terms of APC can partially be due to the similarity between GAP and analysis computation: sup_apc trusts assessors which correctly rank the top systems in the raining topicset, and is then desirable that this property is inherited by the merged measure. Good RMSE results, however, prove that sup_apc and sup_tau measures are accurate not only for the top systems, but for the full set of runs.

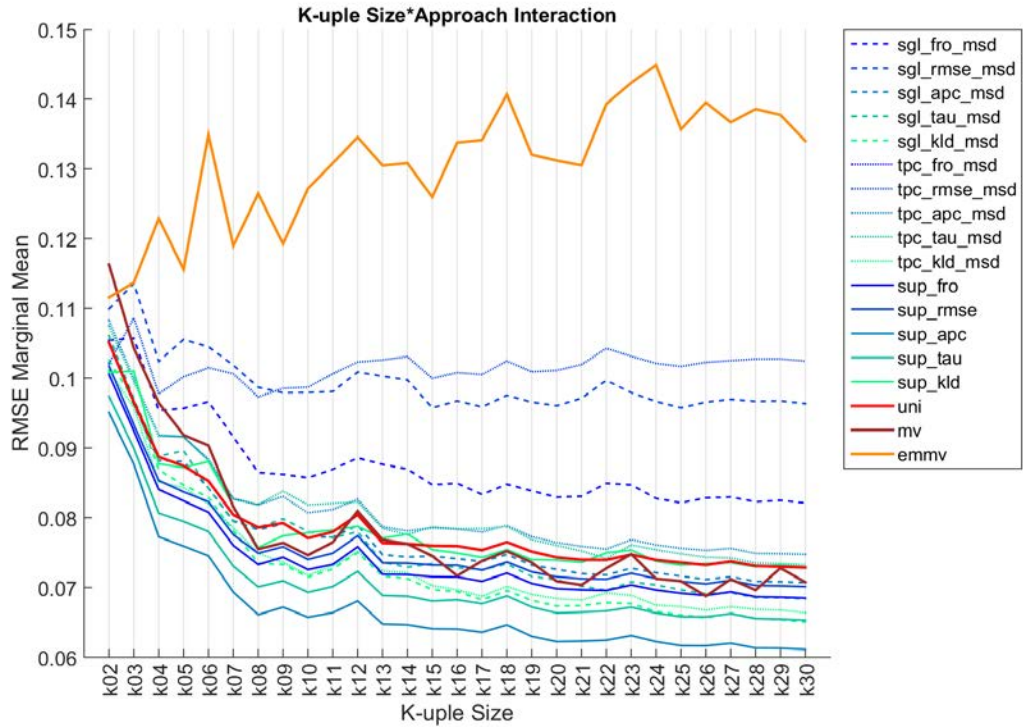


Figure 5.18: RMSE Approach*Kuple interaction effect (5 test topics)

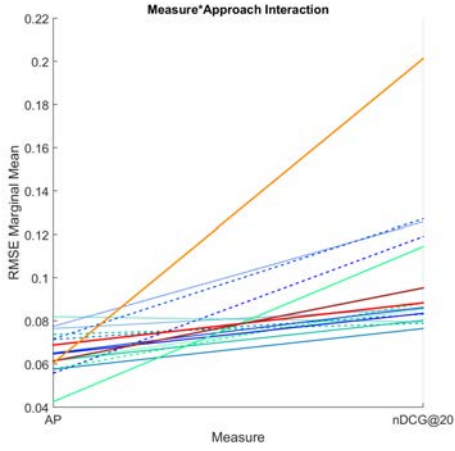


Figure 5.19: RMSE
Approach*Measure interaction effect
(5 test topics)

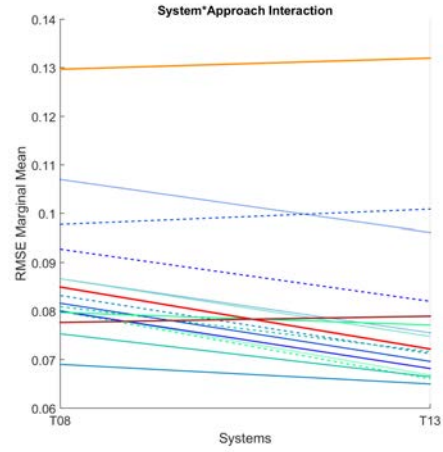


Figure 5.20: RMSE Approach*Runset
interaction effect (5 test topics)

5.3 Results with different topicset sizes

In order to understand how the size of training and test topicsets affects s-AWARE performances, we analyse now the results of the experiments performed using 7 and 3 training topics.

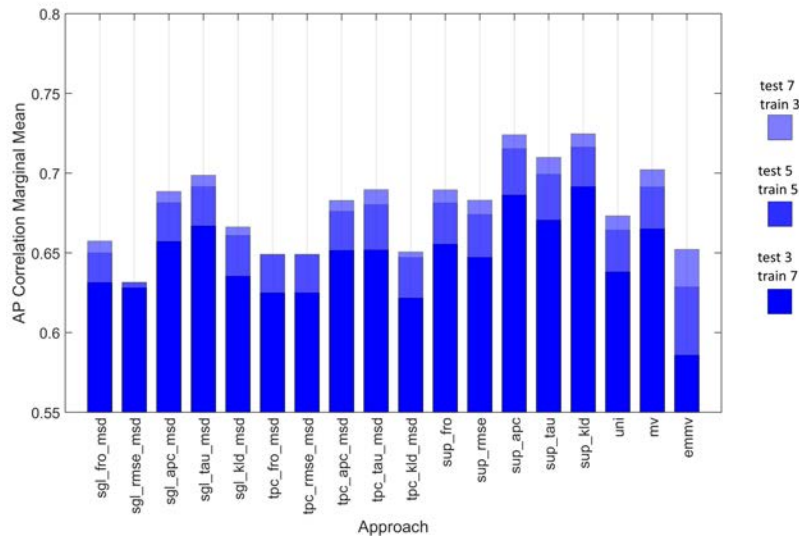


Figure 5.21: AP Correlation of approaches for different topicset sizes

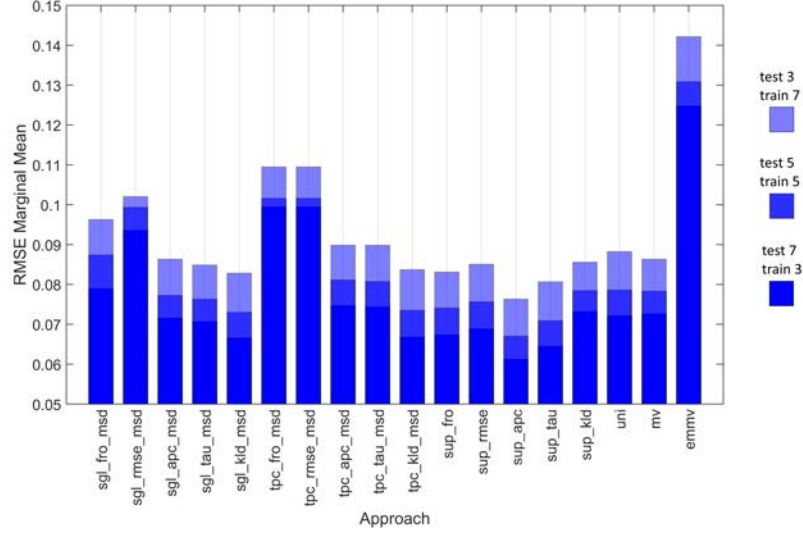


Figure 5.22: RMSE of approaches for different topicset sizes

Figures 5.21 and 5.22 represent respectively the trends of AP Correlation and RMSE between gold measures and approach measures with different topicset size.

Increasing the number of topics used for the test, all the approaches improve their performance in terms of both AP Correlation and RMSE, exploiting the greater quantity of available data for measure merging. While this behaviour is in some way desired and plausible for classic approaches and u-AWARE, we expected s-AWARE performance being negative affected by the smallest dimension of the training set used for the accuracy computation, getting closer to the AWARE-uni performance. Motivated by this results, we further investigated the possible causes for which this doesn't happen, that will be discussed below. ANOVA tables and plots relative to the 3-topicset and 7-topicset experiments follow the same trend of those presented in the previous section: sup_apc and sup_tau still perform better than all the other approaches, expectation maximization improves its performance more than the other approaches, reaching AWARE-uni in terms of AP correlation for big kuples but remaining far from s-AWARE performance. The full set of plots and ANOVA tables from the experiments is reported in Appendix A.

We briefly discuss about the good performances of s-AWARE approaches even with a small training set: to better understand this behaviour we investigated the quality of the assessors' pools. We extracted a small pool taking the first 20

CHAPTER 5. EXPERIMENTAL RESULTS

TOPIC ASSESSOR	411	416	417	420	427	432	438	445	446	447
BUPTPRISZHS	0.9577	0.8601	0.9257	0.9026	0.8792	0.9752	0.8193	0.8916	0.8364	0.9487
NEUEM1	0.9423	0.8497	0.4053	0.3377	0.9101	0.3665	0.8368	0.8940	0.7262	0.4103
NEUEIo2	0.9442	0.8357	0.9221	0.8896	0.9045	0.9752	0.8336	0.8819	0.8381	0.9615
NEUEIo3	0.9481	0.8357	0.9209	0.8896	0.9045	0.9789	0.7702	0.8916	0.8214	0.9637
NEUEIo4	0.9442	0.8357	0.9209	0.8896	0.9045	0.9752	0.7718	0.8916	0.8214	0.9615
NEUEIo5	0.9442	0.8357	0.9221	0.8896	0.9045	0.9752	0.7971	0.8819	0.8381	0.9615
NEUNugget12	0.9442	0.8252	0.9209	0.8896	0.8961	0.9516	0.8082	0.8313	0.7997	0.9637
Orc2G	0.8654	0.4895	0.7998	0.5552	0.6854	0.6994	0.6197	0.6530	0.8314	0.6197
Orc2GUL	0.8538	0.5070	0.7962	0.5519	0.6882	0.6795	0.6149	0.6313	0.7012	0.6175
Orc2GULConf	0.8500	0.5140	0.7938	0.5292	0.6938	0.6820	0.6181	0.6241	0.7012	0.6197
OrcVB1	0.7481	0.5070	0.7614	0.5747	0.7022	0.7404	0.5578	0.5976	0.7295	0.6987
OrcVB1Conf	0.7442	0.5070	0.7614	0.5747	0.7022	0.7391	0.5578	0.6000	0.7295	0.6987
OrcVBW16Conf	0.7385	0.4790	0.7602	0.5974	0.7472	0.6820	0.5959	0.5494	0.6845	0.6560
OrcVBW80	0.7115	0.5979	0.8165	0.6948	0.6517	0.7714	0.6292	0.6265	0.7279	0.7201
OrcVBW80Conf	0.7115	0.5979	0.8165	0.6948	0.6517	0.7714	0.6292	0.6265	0.7279	0.7201
OrcVBW9Conf	0.8115	0.4720	0.7626	0.5000	0.8034	0.7056	0.5737	0.5639	0.6845	0.7009
SSEC3excl	0.9500	0.9790	0.9580	0.9286	0.9719	0.9466	0.9303	0.9614	0.9199	0.9808
SSEC3incl	0.9385	0.9580	0.9412	0.9058	0.9663	0.9354	0.8875	0.9590	0.9149	0.9936
SSEC3inclML	0.9423	0.9685	0.9532	0.9188	0.9719	0.9404	0.8954	0.9614	0.9182	0.9957
SSECML2to99	0.9327	0.9580	0.9412	0.9058	0.9691	0.9354	0.8843	0.9590	0.9098	0.9957
SSECML50pct	0.9365	0.9580	0.9400	0.9026	0.9691	0.9404	0.8859	0.9614	0.9065	0.9957
SSECML75pct	0.9365	0.9615	0.9400	0.9026	0.9691	0.9404	0.8859	0.9614	0.9032	0.9957
SSML2pct	0.9327	0.9615	0.9388	0.9091	0.9579	0.9478	0.8827	0.9590	0.9165	0.9936
SSNoEC	0.9327	0.9615	0.9365	0.9091	0.9635	0.9441	0.8811	0.9639	0.9065	0.9936
UlowaS01r	0.9462	0.7902	0.9089	0.8831	0.8989	0.9776	0.8019	0.8602	0.8013	0.9637
UlowaS02r	0.9462	0.7867	0.9029	0.8701	0.8989	0.9702	0.7892	0.8554	0.7947	0.9637
UlowaS03r	0.9462	0.7902	0.9053	0.8831	0.9017	0.9764	0.8035	0.8651	0.8047	0.9637
yorku12cs01	0.7923	0.6818	0.8765	0.6364	0.7219	0.9255	0.7132	0.6554	0.7596	0.8483
yorku12cs02	0.8038	0.6573	0.8321	0.5974	0.7416	0.8981	0.7496	0.6771	0.7813	0.9338
yorku12cs03	0.7654	0.6329	0.8921	0.5487	0.7022	0.7888	0.6910	0.6892	0.8598	0.8312
yorku12cs04	0.7654	0.6503	0.8909	0.5974	0.7022	0.8050	0.6878	0.6916	0.8581	0.8526

Table 5.3: assessors average agreement with gold standard considering the top 20 documents for each run

retrieved documents from each run of TREC08 and TREC13, and we examined the fraction of agreement between each assessor judgements and the gold standard judgements on each topic. The results of this analysis are shown in table 5.3. We notice that a the most part of the assessors achieve similar agreement scores across most of the topics. This pool agreement drives to similar mean measures when averaging among 3, 5 or 7 topics, even if (from the APC and RMSE results) we can notice that performances slightly improve increasing the number of topics. Similar mean measures lead to similar accuracy scores and then to similar behaviour. This indicates that, almost for this dataset, different sets of topics are not a discriminative factor for assessor accuracy. If we are able to determine a small set of topics on which assessors presumably behave as in the majority of topics, we can then achieve good results with s-AWARE approaches at very low cost.

CONCLUSIONS AND FUTURE WORK

In this thesis we proposed a new supervised approach to exploit crowd assessors relevance judgements for information retrieval evaluation. We proposed this approaches as a new part of the AWARE probabilistic framework, that follow a different methodology with respect to the classic approaches.

AWARE aim to combine multiple assessors merging the evaluation measures computed considering each assessor's judgements as ground truth. This methodology, unlike the classic approaches that aim to create a single ground truth combining assessors' pools, allows to consider the not negligible different impact that mislabelled documents at pool level can lead on different evaluation measures or systems.

S-AWARE approach, combines assessors' measures based to accuracy scores computed with a set of different dissimilarity measures between the gold standard and each assessor: evaluation measures are computed for each retrieval system on a training set of topics, an accuracy score for each assessor is computed to be proportional to the closeness between assessor measures and gold standard measures. In order to consider different ways in which an assessor can be "close" to the gold standard, we developed two approaches based on the real value distance between the measures (named `sup_fro` and `sup_rmse`), one approach based on the comparison between probability distribution of the measures (`sup_kld`) and two approaches based on the comparison between the

system ranking induced by the measures (sup_tau and sup_apc).

To test our approaches we considered as crowd assessors 31 pools submitted to TREC 2012 Crowdsourcing track and we used this pools to evaluate the performance of the runs coming from TREC 08 AdHoc and TREC 13 Robust evaluation campaigns.

The evaluation measures used in our experiments were Average Precision and normalized Discounted Cumulative Gain computed up to rank 20. After the measure computation, we analysed the results using AP correlation to understand how different approaches perform on ranking the systems and RMSE to determine which approach is better in estimating the gold measure values.

We tested our five approaches against:

- Majority Vote and Expectation Maximization, that are two classic and common approaches
- AWARE-uni approach, that uses AWARE methodology with uniform accuracy weights for all the assessors
- u-AWARE approaches, the unsupervised part of the AWARE framework that exploit the same dissimilarity functions to compute accuracy scores proportional to the remoteness of assessors measures from random assessors measures.

We merged together different kuples of 2 to 30 assessors, to investigate how the number of merged assessors affect the approaches performance. Different approaches, measures, systems and kuples constitute a set of factors for which all the combinations are tested. ANOVA analysis is then computed to determine how these factors and combinations of factors influence the behaviour of the approaches.

Experimental results show that measures, approaches and its interaction largely impact on performance, strengthening the motivations behind AWARE methodology.

S-AWARE approaches always perform better than the uniform case and most of s-AWARE approaches behave better than Majority vote, in particular with a small set of merged assessors.

S-AWARE often outperform the corresponding u-AWARE approaches, and

approaches based on ranking dissimilarity usually work better than the others approaches.

We repeated then all the experiments varying the size of test set and training set of topics for s-AWARE approaches.

Results from this further experiments state that some s-AWARE approaches still perform better than the other approaches.

6.1 Future Work

Results presented in this thesis are based on a small set of measures and a small set of topics. Further research should use a larger dataset to validate the results obtained so far, also considering a greater set of evaluation measures. A bigger dataset would allow us also to move to more complex algorithms for accuracy computation.

In our experiments we noticed that some GAPs perform better with a certain evaluation measure than another: a first idea should then be to combine multiple GAPs for different evaluation measures using for each measure the GAP, or the combination of GAPs, which better perform in terms of similarity between assessor and gold measures on the training topics. The aggregation of different GAPs could be done in a similar way to weight computation in u-AWARE approaches.

Another result from the experiments is that s-AWARE accuracy scores are sometimes flattened by the similar values of the measures. In order to better highlight good and poor assessor performance, several techniques can be tested. The first, simple idea is to compute the squared GAP, as done in u-AWARE approaches.

A more complex approach could involve some machine learning techniques: after the s-AWARE training phase, we could perform a validation phase to tune accuracy scores in order to achieve better results. The goal of this process should be finding a local optimum configuration of accuracy scores for which the maximum AP correlation is achieved on a validation set of topics.



PLOTS AND ANOVA TABLES

In this appendix we report plots and tables for the tests performed on topicsets of 3 topics (A.1) and 7 topics (A.2)

A.1 Results with topicset of 3 topics

A.1.1 AP Correlation

	SS	DF	MS	F	p- value	SOA
K-uple Size	1,53301	28	0,05475	289,05128	<0.0001	
Approach	1,29790	17	0,07635	403,06873	<0.0001	0,76600
Measure	0,37522	1	0,37522	1980,96093	<0.0001	0,48672
Systems	0,29149	1	0,29149	1538,90365	<0.0001	0,42414
Approach*Measure	0,41113	17	0,02418	127,67679	<0.0001	0,50772
Approach*Systems	0,081336	17	0,00478	25,25937	<0.0001	0,16493
Measure*Systems	0,41538	1	0,41538	2192,97619	<0.0001	0,51215
Error	0,37978	2005	0,00019			
Total	4,78525	2087				

Table A.1: ANOVA table for AP Correlation (3 test topics)

APPENDIX A. PLOTS AND ANOVA TABLES

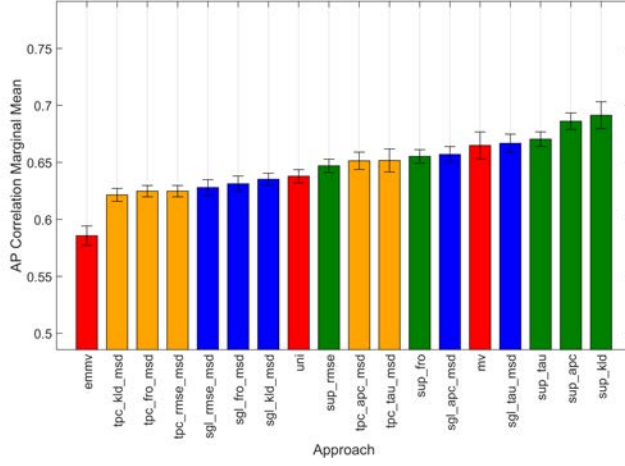


Figure A.1: APC Approach main effect (3 test topics)

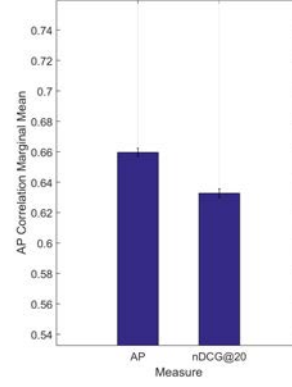


Figure A.2: APC Measure main effect (3 test topics)

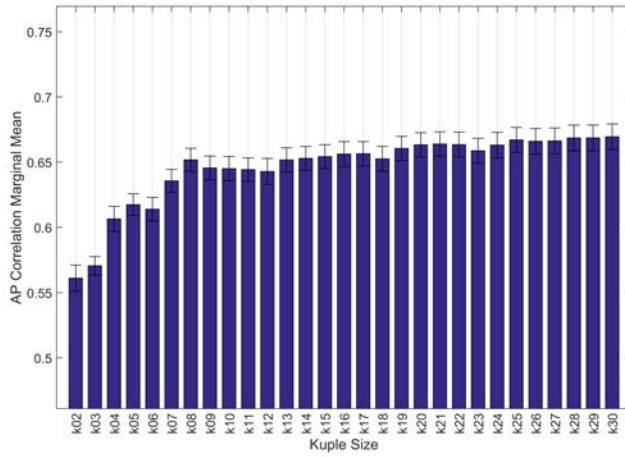


Figure A.3: APC Kupple main effect (3 test topics)

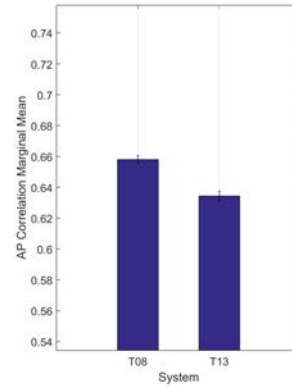


Figure A.4: APC Runset main effect (3 test topics)

A.1. RESULTS WITH TOPICSET OF 3 TOPICS

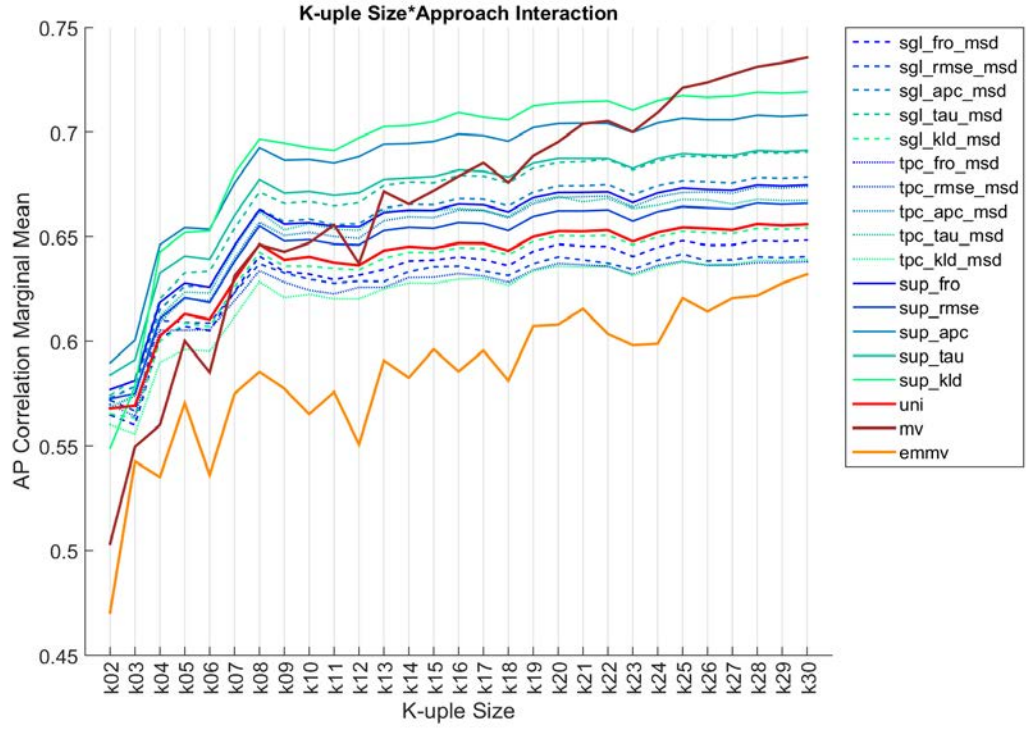


Figure A.5: APC Approach*Kuple interaction effect (3 test topics)

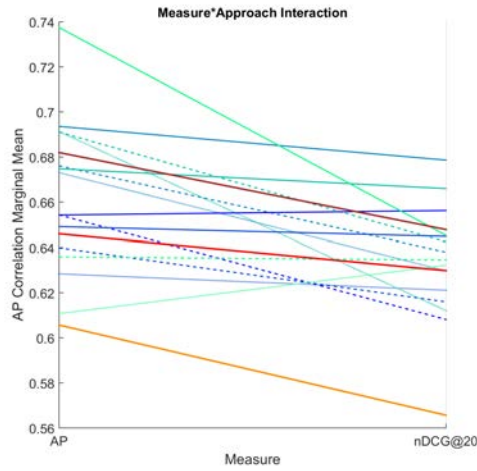


Figure A.6: APC Approach*Measure interaction effect (3 test topics)

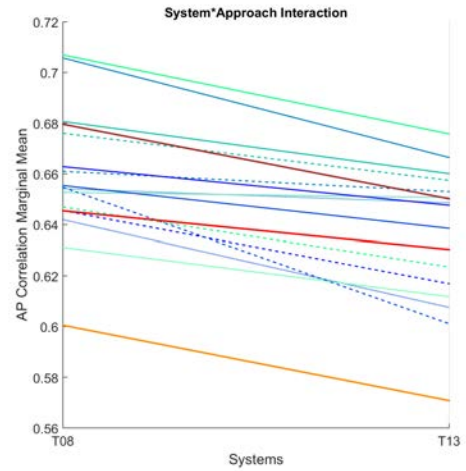


Figure A.7: APC Approach*Runset interaction effect (3 test topics)

A.1.2 RMSE

	SS	DF	MS	F	p- value	SOA
K-uple Size	0,09226	28	0,00330	53,67149	<0.0001	
Approach	0,47308	17	0,02783	453,28833	<0.0001	0,78644
Measure	0,76346	1	0,76346	12435,79244	<0.0001	0,85623
Systems	0,02778	1	0,02778	452,50846	<0.0001	0,17779
Approach*Measure	0,49570	17	0,02916	474,96009	<0.0001	0,79419
Approach*Systems	0,01206	17	0,00071	11,55128	<0.0001	0,07911
Measure*Systems	0,00173	1	0,00173	28,12493	<0.0001	0,01282
Error	0,12309	2005	0,00006			
Total	1,98916	2087				

Table A.2: ANOVA table for RMSE (3 test topics)

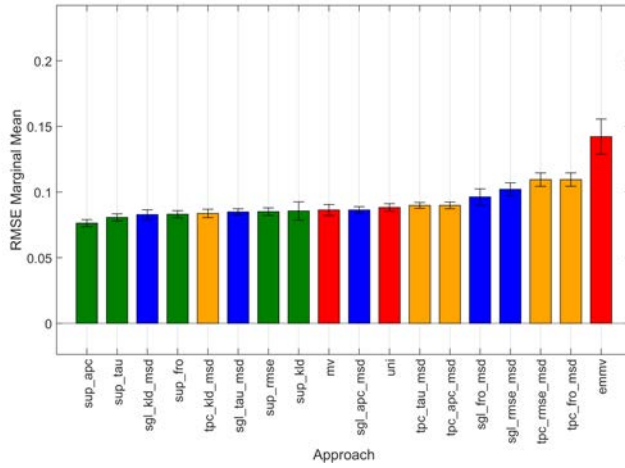


Figure A.8: RMSE Approach main effect (3 test topics)

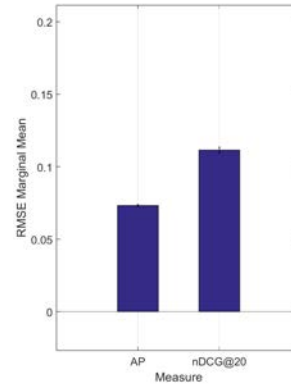


Figure A.9: RMSE Measure main effect (3 test topics)

A.1. RESULTS WITH TOPICSET OF 3 TOPICS

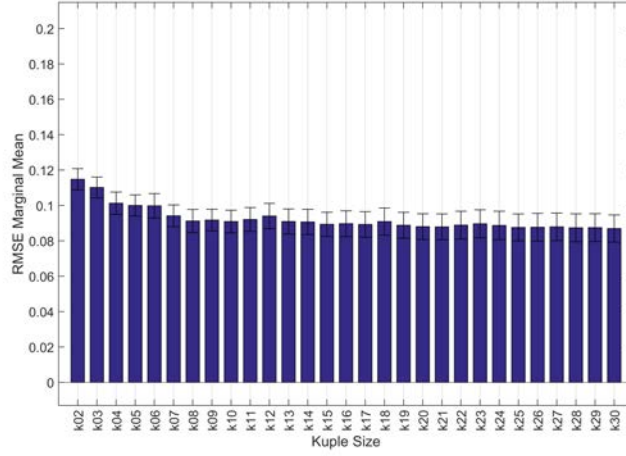


Figure A.10: RMSE Kuple main effect (3 test topics)

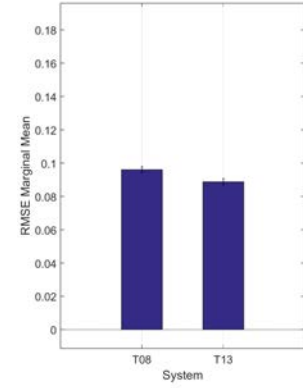


Figure A.11: RMSE Runset main effect (3 test topics)

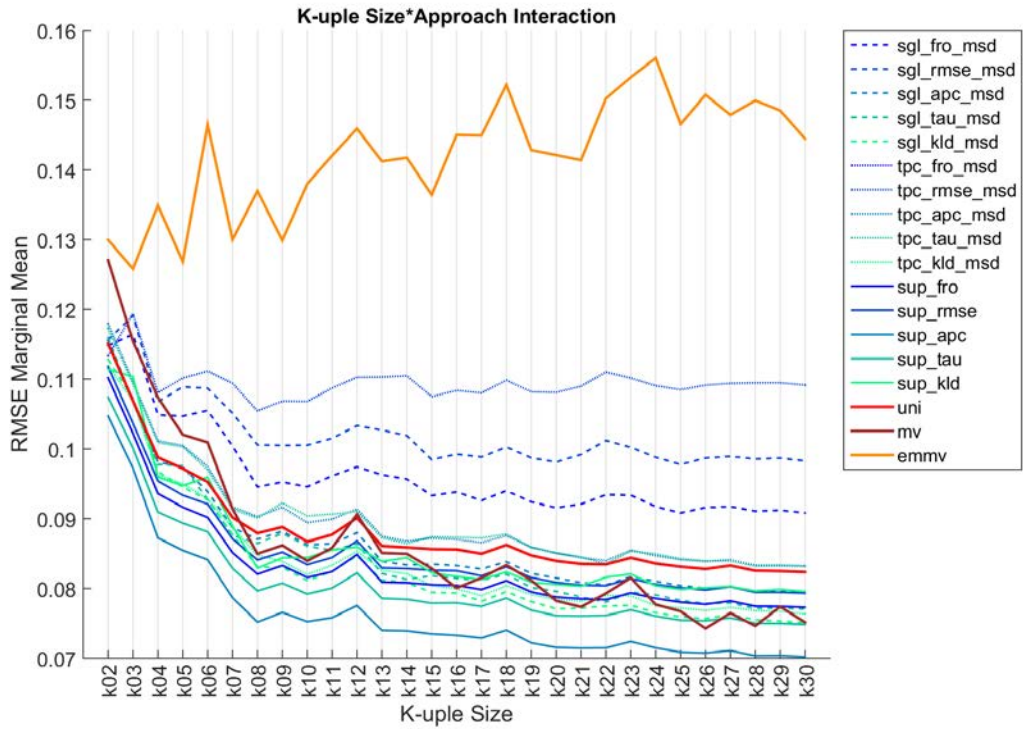


Figure A.12: RMSE Approach*Kuple interaction effect (3 test topics)

APPENDIX A. PLOTS AND ANOVA TABLES

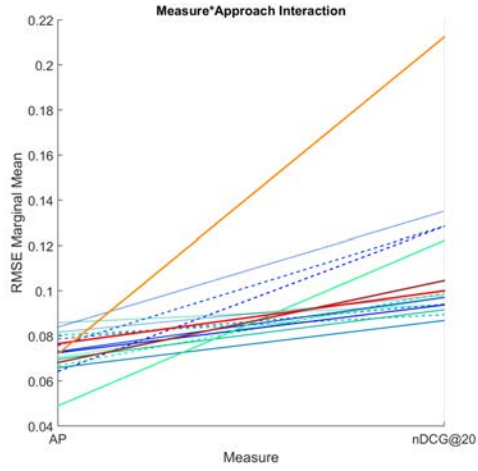


Figure A.13: RMSE Approach*Measure interaction effect (3 test topics)

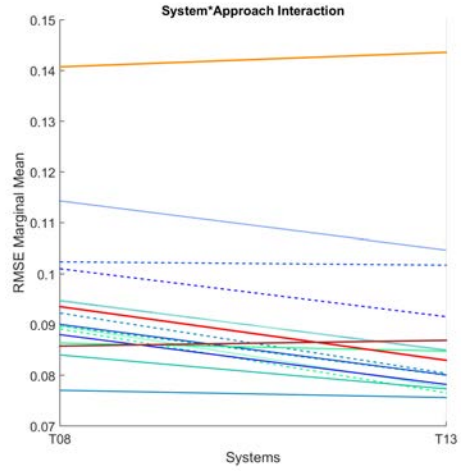


Figure A.14: RMSE Approach*Runset interaction effect (3 test topics)

A.2 Results with topicset of 7 topics

A.2.1 AP Correlation

	SS	DF	MS	F	p- value	SOA
K-uple Size	1,05321	28	0,03761	151,00729	<0.0001	
Approach	1,45821	17	0,08578	344,35860	<0.0001	0,73653
Measure	0,66655	1	0,66655	2675,94252	<0.0001	0,56162
Systems	0,07361	1	0,07361	295,50691	<0.0001	0,12361
Approach*Measure	0,26757	17	0,01574	63,18643	<0.0001	0,33612
Approach*Systems	0,20151	17	0,01185	47,58593	<0.0001	0,27499
Measure*Systems	0,94652	1	0,94652	3799,90612	<0.0001	0,64531
Error	0,49943	2005	0,00025			
Total	5,16660	2087				

Table A.3: ANOVA table for AP Correlation (7 test topics)

A.2. RESULTS WITH TOPICSET OF 7 TOPICS

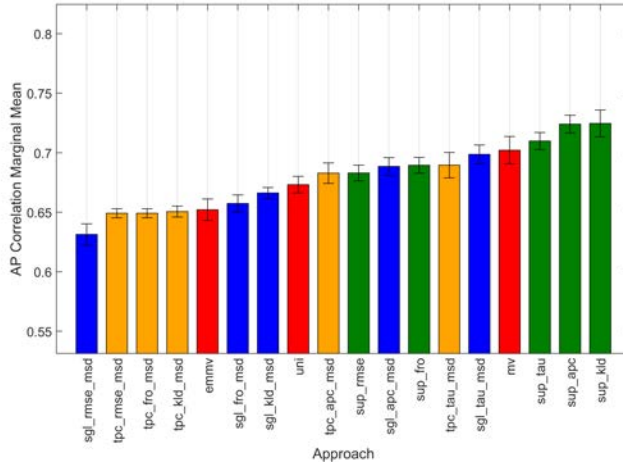


Figure A.15: APC Approach main effect (7 test topics)

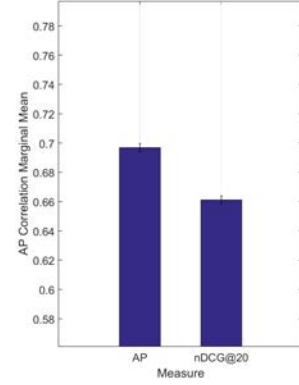


Figure A.16: APC Measure main effect (7 test topics)

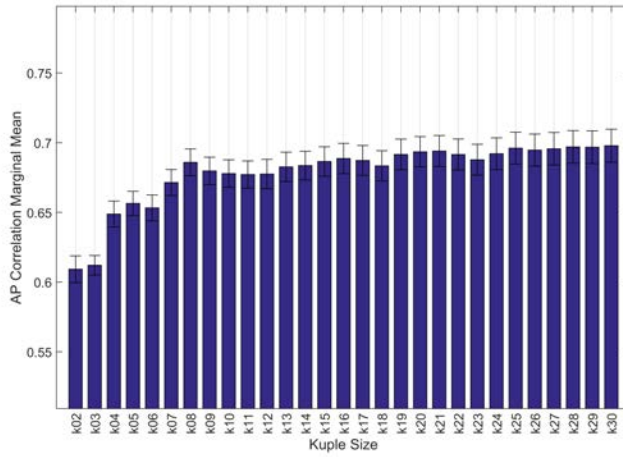


Figure A.17: APC Kuplet main effect (7 test topics)

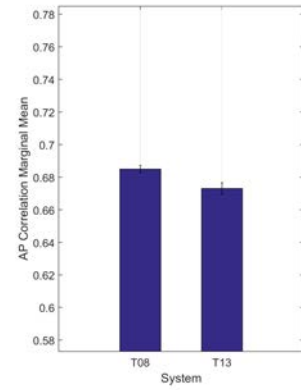


Figure A.18: APC Runset main effect (7 test topics)

APPENDIX A. PLOTS AND ANOVA TABLES

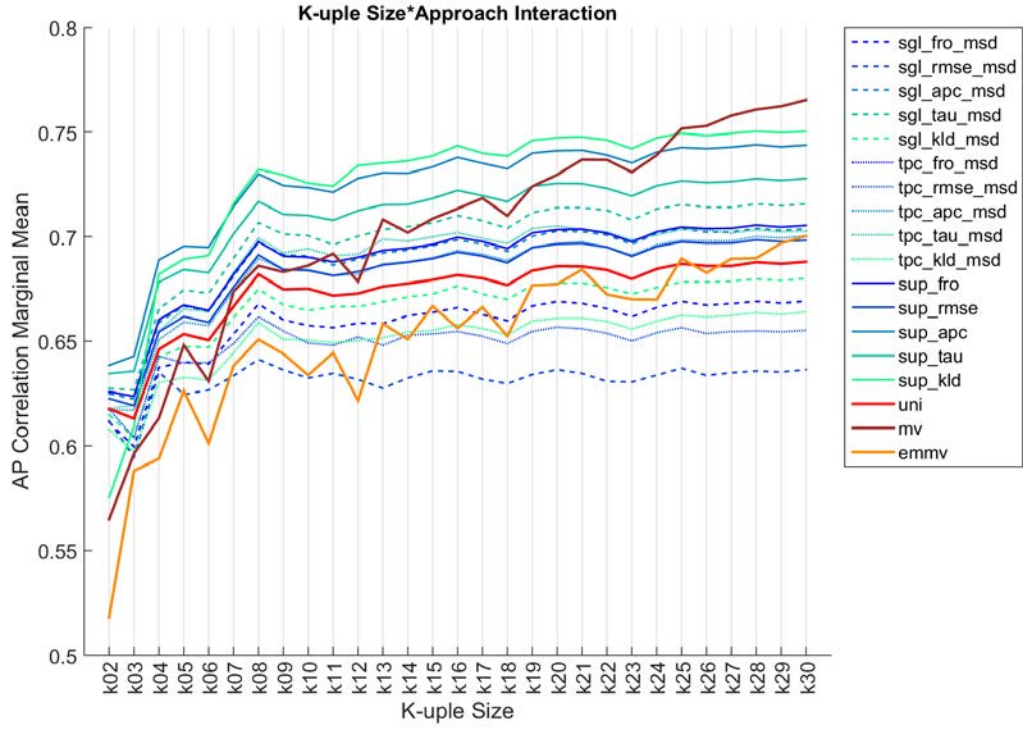


Figure A.19: APC Approach*Kuple interaction effect (7 test topics)

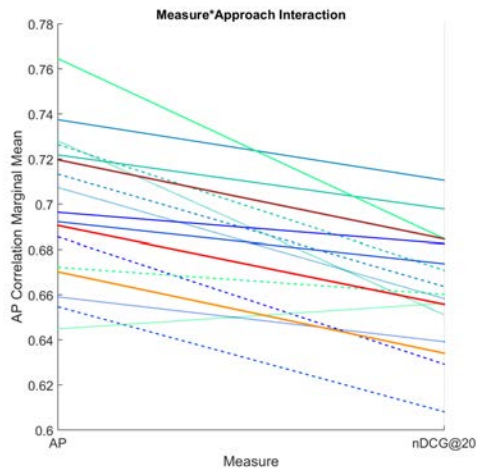


Figure A.20: APC Approach*Measure interaction effect (7 test topics)

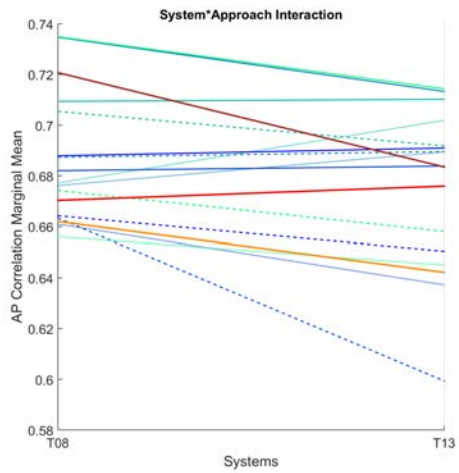


Figure A.21: APC Approach*Runset interaction effect (7 test topics)

A.2.2 RMSE

	SS	DF	MS	F	p- value	SOA
K-uple Size	0,06799	28	0,00243	32,06742	<0.0001	
Approach	0,51858	17	0,03050	402,82432	<0.0001	0,76589
Measure	0,68148	1	0,68148	8999,23970	<0.0001	0,81166
Systems	0,03675	1	0,03675	485,29366	<0.0001	0,18827
Approach*Measure	0,58751	17	0,03456	456,37173	<0.0001	0,78757
Approach*Systems	0,01922	17	0,00113	14,92999	<0.0001	0,10186
Measure*Systems	0,00566	1	0,00566	74,75456	<0.0001	0,03412
Error	0,15183	2005	0,00008			
Total	2,06903	2087				

Table A.4: ANOVA table for RMSE (7 test topics)

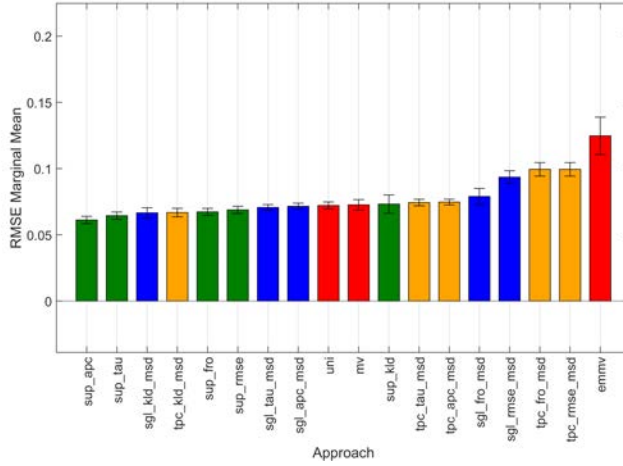


Figure A.22: RMSE Approach main effect (7 test topics)

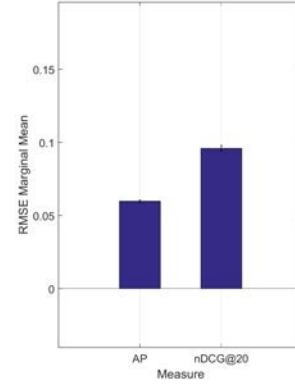


Figure A.23: RMSE Measure main effect (7 test topics)

APPENDIX A. PLOTS AND ANOVA TABLES

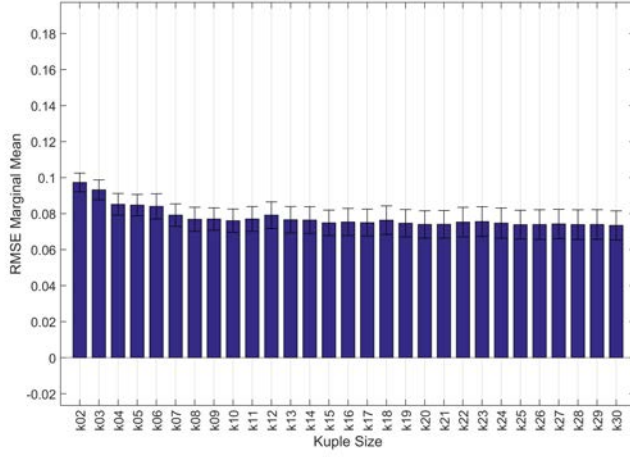


Figure A.24: RMSE Kupple main effect (7 test topics)

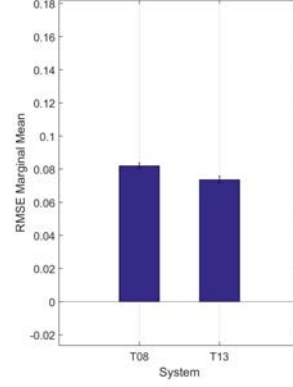


Figure A.25: RMSE
Runset main effect (7 test
topics)

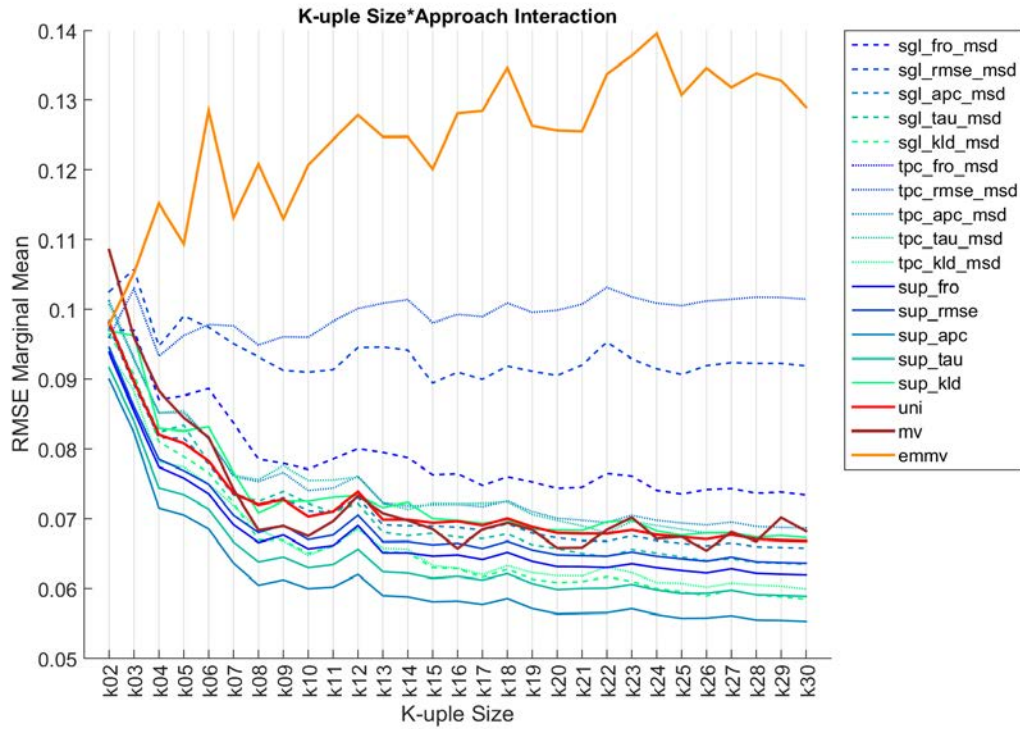


Figure A.26: RMSE Approach*Kupple interaction effect (7 test topics)

A.2. RESULTS WITH TOPICSET OF 7 TOPICS

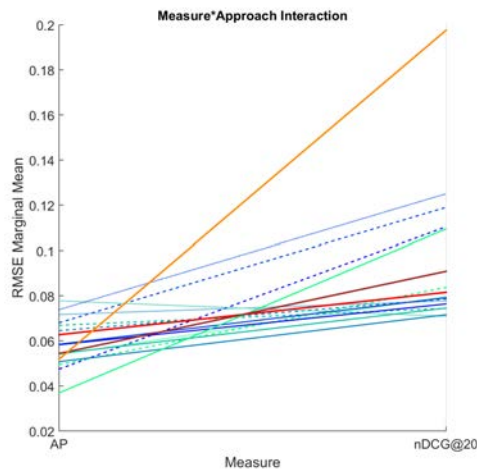


Figure A.27: RMSE Approach*Measure interaction effect (7 test topics)

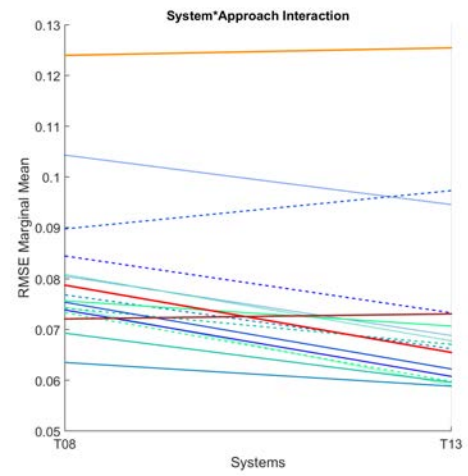


Figure A.28: RMSE Approach*Runset interaction effect (7 test topics)

BIBLIOGRAPHY

- [1] D. Harman, “Is the cranfield paradigm outdated?,” in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, p. 1, ACM Press, 2010.
- [2] M. Hosseini, I. J. Cox, N. Milić-Frayling, G. Kazai, and V. Vinay, “On aggregating labels from multiple crowd workers to infer relevance of documents,” in *Lecture Notes in Computer Science*, pp. 182–194, Springer Berlin Heidelberg, 2012.
- [3] M. Hosseini, I. J. Cox, N. Milić-Frayling, G. Kazai, and V. Vinay, “On aggregating labels from multiple crowd workers to infer relevance of documents,” in *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR’12, pp. 182–194, 2012.
- [4] M. Bashir, J. Anderton, J. Wu, M. Ekstrand-Abueg, P. B. Golbus, V. Pavlu, and J. A. Aslam, “Northeastern university runs at the trec12 crowdsourcing track,” in *Proceedings of the TREC 2012 crowdsourcing track* (E. M. Voorhees and L. P. Buckland, eds.), pp. 1–11, National Institute of Standards and Technology (NIST), 2012.
- [5] W. Tang and M. Lease, “Semi-supervised consensus labeling for crowdsourcing,” in *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Information Retrieval, 2011*, pp. 36–41, jul 2011.
- [6] T. Tian, J. Zhu, and Y. Qiaoben, “Max-margin majority voting for learning from crowds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2480–2494, oct 2019.
- [7] S. Whiting, J. Perez, G. Zuccon, T. Leelanupab, and J. Jose, “University of glasgow (qirdcsuog) at trec crowdsourcing 2001: Turkrank – network

- based worker ranking in crowdsourcing,” in *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*, pp. 1–7, jan 2011.
- [8] R. Nellapati, S. Peerreddy, and P. Singhal, “Skierarchy: Extending the power of crowdsourcing using a hierarchy of domain experts, crowd and machine learning,” in *Proceedings of the TREC 2012 crowdsourcing track*, pp. 1–11, 2012.
- [9] M. Ferrante, N. Ferro, and M. Maistro, “AWARE: Exploiting Evaluation Measures to Combine Multiple Assessors,” *ACM Transactions on Information Systems*, vol. 36, pp. 1–38, aug 2017.
- [10] B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*.
USA: Addison-Wesley Publishing Company, 1st ed., 2009.
- [11] K. S. Jones, “The cranfield tests,” in *Information retrieval experiment*, ch. 14, pp. 256–284, Butterworth & Co Ltd, 1981.
- [12] M. Angelini, N. Ferro, G. Santucci, and G. Silvello, “VIRTUE: A visual tool for information retrieval performance evaluation and failure analysis,” *Journal of Visual Languages & Computing*, vol. 25, pp. 394–413, aug 2014.
- [13] M. P. Ekstrand-Abueg, *A comprehensive method for automating test collection creation and evaluation for retrieval and summarization systems*.
PhD thesis, Boston, Massachusetts, Northeastern University, 2016.
- [14] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. LIX, pp. 433–460, oct 1950.
- [15] A. J. Quinn and B. B. Bederson, “Human computation,” in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, pp. 1403–1412, ACM Press, 2011.
- [16] L. Von Ahn, *Human Computation*.
PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.

- [17] J. Howe, “The rise of crowdsourcing,” *Wired Magazine*, vol. 14, p. 2, jan 2006.
- [18] J. Surowiecki, *The Wisdom of the crowds*. Anchor Books, 2004.
- [19] D. C. Brabham, *Crowdsourcing*. The MIT Press, 2013.
- [20] O. Alonso, *The Practice of Crowdsourcing*. Morgan & Claypool Publishers, 2019.
- [21] J. D. Orkin, *Collective artificial intelligence : simulated role-playing from crowdsourced data*. PhD thesis, Massachussets institute of technology, 2013.
- [22] B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas, “Crowd-sourcing for multiple-choice question answering,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI’14, pp. 2946–2953, AAAI Press, 2014.
- [23] A. Kovashka, O. Russakovsky, L. Fei-Fei, and K. Grauman, “Crowdsourcing in computer vision,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 10, no. 3, pp. 177–243, 2016.
- [24] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “reCAPTCHA: Human-based character recognition via web security measures,” *Science*, vol. 321, pp. 1465–1468, aug 2008.
- [25] D. Difallah, E. Filatova, and P. Ipeirotis, “Demographics and dynamics of mechanical turk workers,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM ’18*, pp. 135–143, ACM Press, 2018.
- [26] L. von Ahn, “Games with a purpose,” *Computer*, vol. 39, pp. 92–94, jun 2006.
- [27] L. von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proceedings of the 2004 conference on Human factors in computing systems - CHI ’04*, pp. 319–326, ACM Press, 2004.

- [28] L. von Ahn, R. Liu, and M. Blum, “Peekaboomb: a game for locating objects in images,” in *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems '06*, pp. 55–64, ACM Press, 2006.
- [29] C. Eickhoff, C. G. Harris, P. Srinivasan, and A. P. de Vries, “GEAnn - Games for Engaging Annotations,” in *Proceedings of the ACM SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR 2011)* (M. Lease, V. Hester, A. Sorokin, and E. Yilmaz, eds.), (Beijing, China), p. 63, July 2011.
- [30] L. von Ahn, M. Kedia, and M. Blum, “Verbosity: a game for collecting common-sense facts,” in *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, pp. 75–78, ACM Press, 2006.
- [31] H. Ma, R. Chandrasekar, C. Quirk, and A. Gupta, “Improving search engines using human computation games,” in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, pp. 275–284, ACM, 2009.
- [32] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, and Z. Popović, “Predicting protein structures with a multiplayer online game,” *Nature*, vol. 466, pp. 756–760, aug 2010.
- [33] T.-Y. Liu, “Applications of learning to rank,” in *Learning to Rank for Information Retrieval*, pp. 181–191, Springer Berlin Heidelberg, 2011.
- [34] T. Yan, V. Kumar, and D. Ganesan, “Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, pp. 77–90, ACM Press, 10 2010.
- [35] O. Alonso and S. Mizzaro, “Using crowdsourcing for trec relevance assessment,” *Inf. Process. Manage.*, vol. 48, pp. 1053–1066, Nov. 2012.
- [36] O. Alonso, D. E. Rose, and B. Stewart, “Crowdsourcing for relevance evaluation,” *ACM SIGIR Forum*, vol. 42, pp. 9–15, nov 2008.

- [37] W. Webber and J. Pickens, “Assessor disagreement and text classifier accuracy,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, pp. 929–932, ACM Press, 2013.
- [38] D. Goldberg, A. Trotman, X. Wang, W. Min, and Z. Wan, “Further insights on drawing sound conclusions from noisy judgments,” *ACM Transactions on Information Systems*, vol. 36, pp. 1–31, apr 2018.
- [39] P. Clough, M. Sanderson, J. Tang, T. Gollins, and A. Warner, “Examining the limits of crowdsourcing for relevance assessment,” *IEEE Internet Computing*, vol. 17, pp. 32–38, jul 2013.
- [40] M. Smucker and C. P. Jethani, “The Crowd vs. the Lab: A Comparison of Crowd-Sourced and University Laboratory Participant Behavior,” in *Proceedings of the ACM SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR 2011)* (M. Lease, V. Hester, A. Sorokin, and E. Yilmaz, eds.), (Beijing, China), pp. 9–14, July 2011.
- [41] M. D. Smucker and C. P. Jethani, “Measuring assessor accuracy: a comparison of nist assessors and user study participants,” in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, pp. 1231–1232, ACM Press, 2011.
- [42] S. Wakeling, M. Halvey, R. Villa, and L. Hasler, “A comparison of primary and secondary relevance judgements for real-life topics,” in *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval - CHIIR '16*, pp. 173–182, ACM Press, 2016.
- [43] A. Aker, M. El-Haj, M.-D. Albakour, and U. Kruschwitz, “Assessing crowdsourcing quality through objective tasks,” in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, eds.), pp. 1456–1461, European Language Resources Association (ELRA), may 2012.

- [44] M. Kutlu, T. McDonnell, Y. Barkallah, T. Elsayed, and M. Lease, “Crowd vs. expert: What can relevance judgment rationales teach us about assessor disagreement?,” in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18*, pp. 805–814, ACM Press, jun 2018.
- [45] J. Vuurens, A. de Vries, and C. Eickhoff, “How much spam can you take? an analysis of crowdsourcing results to increase accuracy,” in *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Information Retrieval, 2011*, pp. 48–55, jul 2011.
- [46] J. B. Vuurens and A. P. de Vries, “Obtaining high-quality relevance judgments using crowdsourcing,” *IEEE Internet Computing*, vol. 16, pp. 20–27, sep 2012.
- [47] M. Lease, “On quality control and machine learning in crowdsourcing,” in *Proceedings of the 11th AAAI Conference on Human Computation, AAAIWS'11-11*, pp. 97–102, AAAI Press, 2011.
- [48] D. Tao, J. Cheng, Z. Yu, K. Yue, and L. Wang, “Domain-weighted majority voting for crowdsourcing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 163–174, jan 2019.
- [49] H. Li and B. Yu, “Error rate bounds and iterative weighted majority voting for crowdsourcing,” *ArXiv*, nov 2014.
- [50] M. Ferrante, N. Ferro, and E. Losiouk, “Stochastic relevance for crowdsourcing,” in *Proceedings of the 41st European Conference on IR Research*, pp. 755–762, jan 2019.
- [51] S. White and P. Smyth, “Algorithms for estimating relative importance in networks,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, pp. 266–275, ACM Press, 2003.
- [52] M. D. Smucker, G. Kazai, and M. Lease, “Overview of the trec 2012 crowdsourcing track,” in *Proceedings of the 21st Text REtrieval Conference*, National Institute of Standards and Technology (NIST), 2012.

- [53] D. Hawking, E. Voorhees, N. Cranswell, and P. Bailey, “Overview of the trec-8 web track,” in *Proceedings of the 8th Text REtrieval Conference*, National Institute of Standards and Technology (NIST), 2000.
- [54] E. Voorhees, “Overview of the trec 2004 robust retrieval track,” in *Proceedings of the 13th Text REtrieval Conference*, National Institute of Standards and Technology (NIST), 2005.
- [55] A. Rutherford, *ANOVA and ANCOVA: A GLM Approach*. John Wiley & Sons, 2nd ed., 2011.
- [56] N. Ferro and G. Silvello, “A general linear mixed models approach to study system component effects,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*, pp. 25–34, ACM Press, 2016.

RINGRAZIAMENTI

Al termine di questo percorso voglio ringraziare il mio relatore, Prof. Nicola Ferro, per la disponibilità e la pazienza con cui mi ha seguito durante lo sviluppo della mia tesi.

Ringrazio i tanti compagni di viaggio con i quali ho condiviso i miei studi: camminare assieme ha reso questa esperienza più bella. Ringrazio in modo particolare i professori e i dottorandi del gruppo di ricerca IMS, per avermi accolto in laboratorio come uno di loro, consigliandomi e accompagnandomi in un ambiente a me nuovo.

Ringrazio tutta la mia famiglia per avermi dato la possibilità di percorrere questa strada, sostenendomi dall'inizio alla fine.

Ringrazio gli amici tutti, per esserci stati sempre e aver condiviso con me le esperienze più diverse tra divertimento, formazione, lavoro e volontariato: riuscire a conciliare lo studio con altre esperienze è stato per me importante e mi ha reso una persona migliore.

Grazie davvero, perchè anche se questa laurea può sembrare solo un mio risultato, io sono certo che senza di voi non ce l'avrei fatta.

