We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Mining Numbers in Text: A Survey

Minoru Yoshida and Kenji Kita

Abstract

Both words and numerals are tokens found in almost all documents but they have different properties. However, relatively little attention has been paid in numerals found in texts and many systems treated the numbers found in the document in ad-hoc ways, such as regarded them as mere strings in the same way as words, normalized them to zeros, or simply ignored them. Recent growth of natural language processing (NLP) research areas has change this situations and more and more attentions have been paid to the numeracy in documents. In this survey, we provide a quick overview of the history and recent advances of the research of mining such relations between numerals and words found in text data.

Keywords: text mining, numeracy, survey, embedding, natural language inference

1. Introduction

Natural language processing (NLP) is a research field to make machines understand the meaning of a text data, which is a typically a list of words. In some cases, texts are not understandable in their closed form, i.e., without understanding the data other than the words. Numerals are an important form of data in such non-word data not only because many documents are accompanied with related metadata such as publish dates expressed in the form of numbers, but also because the document themselves contain numerals such as "three people", "500 dollars", and "90 cm."

Jointly mining texts and their associated numerical metadata has many variations and many studies have been proposed. For example, predicting stars given with product review texts is typical task of such research areas. Location-aware text mining can be considered as mining of association rules between words and positional data (i.e., longitude and latitude). Even joint learning of texts and images can be seen as mining relations between texts and associated RGB data.

In contrast to such *grounding*-type research, studies on mining numerals explicitly written in text have been getting little attention. However, recently more and more research studies are proposed on this area partly due to the recent advance of deep neural network-based language modeling.

In this survey, we try to provide a quick overview of the history and recent advances of this research field ranging from traditional tasks like information retrieval to emerging ones such as numerical reading comprehension.

2. Traditional tasks

Firstly, we give a survey on the systems which consider treatment of numbers for the traditional tasks such as information retrieval (IR), question answering (QA), and information extraction (IE). Some of the questions or queries of these tasks require the answers to be numbers, hence requiring appropriate treatment of numbers found in the target text.

2.1 Question answering

Question Answering (QA) is a task to find appropriate answers from text to the questions given also in text. Because many type of questions require the answers to be numbers e.g., 8,848 (8,849) (meters) is the answer of the question "how tall is Everest?", some existing QA systems treat numbers appropriately, typically in ad-hoc heuristical ways.

For example, IBM's PIQUANT system for TREC2003 [1] have had the sanity checking module, which use the Cyc knowledge base to check the given answer is valid intervals found in Cyc, e.g., rejecting "200 miles" for the questions for height of a mountain, having the knowledge "mountains are between 1,000 and 30,000 high" from Cyc. Moriceau [2] consider more complicated situation where several numeric answers can be extracted from different Web pages in QA system. They proposed a way to integrate considering the nature of numbers such as number approximations.

2.2 Information retrieval

Similarly to QA tasks, some Information Retrieval (IR) systems return the direct "answers" to the query. Therefore, appropriate treatment of numbers is required for some type of queries. For example, Banerjee et al. [3] introduced Quantity Consensus Queries (QCQs), the answers for which is the quantity intervals, such as "driving time from Paris to Nice". Their proposed algorithm propose and rank intervals considering whether returned snippets is included in the intervals or not. Sarawagi and Chakrabarti [4] proposed a system to answer quantity queries on Web tables such as "escape velocity jupiter." Their system contain the modules to interpret the numbers presented in the table cells to improve the accuracy.

On the contrary, queries also can be numbers. Yoshida et al. [5] proposed a suffix array-based text mining system enhanced with treatment of numbers, which accept range queries like "[1,000 - 10,000] ft'.

2.3 Information extraction

Information extraction (IE) is another type of systems that return the answers to the questions, but in this case the questions are given a priori such as "extract all dates and places of events found in the given documents." Many extracted information is in numerals, so special treatment of numbers often contributes to the improvement of the performance of IE systems.

For example, Bakalov and Fuxman [6] proposed a system to extract numerical attributes of objects given attribute names, seed entities, and related Web pages and properly distinguish the attributes having similar values.

Table 1 summarizes these systems.

System	Special Treatment for Numerals
Question Answering Systems	
IBM's PIQUANT [1]	Matching numerals with values in knowledge base
Moriceau [2]	Integrating multiple numerals
Information Retrieval Systems	
Banerjee et al. [3]	Number interval estimation for a collection of numerals
Sarawagi and Chakrabarti [4]	Understanding numerals in tables
Yoshida et al. [5]	Number range queries for text mining
Information Extraction Systems	
Bakalov and Fuxman [6]	Distinguish similar but different numerals

Table 1.
Systems for Traditional Tasks Considering Numerals.

3. Numerical common sense acquisition

Numerical common senses acquisition is a task to obtain numerical common senses, e.g., the height of mountains have a typical values "1,000 - 10,000 meters". Many numerals found on text typically describe some *attributes* of *objects* such as "25 C" for the temperature of some city, "170 cm" for the height of some person, etc. Obtaining such numerical common senses can contribute to improving various kinds of systems, e.g., anomaly detection or dialogue systems, etc.

We introduce two type of tasks in this type of research. One is a task to directly extract the common senses, and the other is a task to acquire such knowledge as language model parameters.

3.1 Pattern-based extraction of numerical common senses

In this task, the input is a large collection of texts.[1] The output is a database for "typical values" of something.[2]

Typical methods for this task is to use pattern matching to obtain numerals for each attribute described in the given text. For example, the value "80" can be extracted from the sentence "The size of the dog is 80 cm." using the pattern "the size of the A is # cm."

3.1.1 Previous methods

Aramaki et al. [7] proposed to obtain physical size of entities by using Web search with patterns like "book (*cm x *cm)". Bagherinezhad et al. [8] proposed to use knowledge obtained using these patterns with object detection from images to achieve more reliable object size knowledge. Davidov and Rappoport [9] proposed similar approach but augment their method by obtaining terms similar to given object using the Web and WordNet. Takamura and Tsujii [10] took similar approach by using Web search for linguistic patterns e.g., "the size of A", but they enhanced their patterns with more indirect clues such as WordNet relations,

---

[1] It includes the case where the system uses Web search engines where the huge amount of texts are behind the search engines.

[2] It is an attribute of object in most cases.

n-gram corpus for the explicit patterns, e.g., "A is longer than B", and implicit patterns, e.g., "put A in B", through a machine learning approach to determine their weights.

Narisawa et al. [11] proposed to obtain numerical common sense by searching numerical expressions in Web corpus, and calculating distribution of numbers given contexts that are given syntactically such as "verb=give, subj=he, ..." and predict labels for given numbers in text, such as *small*, *normal*, *large*.

Recently, a large dataset called Distribution over Quantities (DoQ), was provided by Elazar et al. [12]. It contains ten dimensions (TIME, CURRENCY, LENGTH, AREA, VOLUME, MASS, TEMPERATURE, DURATION SPEED, VOLTAGE) for various kinds of words including nouns, adjectives, and verbs. They explored co-occurrence of words and numeracy in large Web data.

Table 2 summarizes these approaches and **Table 3** shows the existing data sets.

3.2 Prediction of numbers in sentences

Some researchers tried to acquire numerical common senses as parameters of language modeling. In this type of research, the system directly predicts numbers to fill in the blanks in texts, or assessing feasibility of the number presented in text, without explicitly collecting above-mentioned knowledge bases.

3.2.1 Task definition

In this task, the input is a sentence, or document, where the position for a numeral is masked. The system then outputs a likely value for the masked position. For example, given the sentence "my five-year-old son is [MASK] cm tall.", the

System	Example Patterns	Source
Aramaki et al. [7]	"A (*cm x *cm)"	Web search
Bagherinezhad et al. [8]	"A * x * cm"	Web search, Flickr
	"A is * cm tall"	
	Objects in images	
Davidov and Rappoport [9]	"A is * cm tall"	Web Search, TREC
Takamura and Tsujii [10]	"the size of A",	Web search
	"A is longer than B",	
	WordNet	
Narisawa et al. [11]	Syntactic Patterns	Web corpus
Elazar et al. [12]	co-occurrence	Web corpus

Table 2.
Systems for Numerical Common Sense Acquisition.

Data	Source	Method
DoQ [12]	Web corpus	Collecting co-occurred numerals for each word

Table 3.
Data set for Numerical Common Sense Acquisition.

system is required to answer the likely value to be filled in the position of "[MASK]."

Because the input is a sequence of words, encoder-decoder models are applicable to this task. Especially, the BERT language model is a good match for this problem. BERT is a deep neural network model that consists of modules called *Transformers*. It is trained on the task where the input is a sequence of words with special "[MASK]" tokens, and one of the output is the estimated original word for the position of "[MASK]".

3.2.2 Previous approaches

Several BERT models pretrained on a huge size of text data are available to the public. Using such pretrained language model to predict or assess the numeracy in documents is an emerging trend. Typically, the models are enhanced with the ability to predict numbers by, simply using masked language models by replace the numbers to be predicted with [MASK] tokens, or by adding numeracy inference modules into language models or by fine-tuning setting where output is a *discretized* versions of target numeracy.

Zhang et al. [13] investigated how pretrained language model like BERT can predict (the *discretized* version of) the attribute with continuous numeric values such as MASS or PRICE with evaluation with DoQ. Chen et al. [14] proposed a task of predicting the magnitude of hidden numerals in text and provided a large dataset called Numeracy-600 K. They also reported CNN and RNN-based models to solve this task. Berg-Kirpatrick and Spokoyny [15] proposed more advanced model using BERT and reported that using BERT was better than other models including BiGRU.

On the other hand, Lin et al. [16] considered more difficult task to predict *accurate* number to be filled in the blank in text, like "A bird usually has [MASK] legs". They reported that the current pretrained models including BERT and RoBERTa performed poorly.

A language model that did not use encoder-decoder model was also proposed. Spithourakis and Riedel [17] proposed a language model for a sequence of words and numerals, which gives the probability for words and numerals simultaneously. For example, it gives the probability of the numeral "50,000" appearing just after the word sequence "the number of video-game consoles I have is". They introduced the probabilities of being words or numerals for each token, and modeled the probability for numerals independently of that of words, using some variations including digit-based RNN and mixture of Gaussians.

Table 4 summarizes the approaches and **Table 5** shows the dataset for this task.

System	Model	Task
Zhang et al. [13]	BERT fine tuning	Magnitude[a] prediction
Chen et al. [14]	CNN, RNN	Magnitude prediction
Berg-Kirpatrick and Spokoyny [15]	BERT	Magnitude prediction
Lin et al. [16]	BERT, RoBERTa	Accurate number prediction
Spithourakis and Riedel [17]	generative probability	Perplexity maximization
	(digit-based RNN and Gaussian)	

[a]*Discretized (binned) values, e.g., categorized by the number of digits of numerals.*

Table 4.
Number Prediction Data Sets.

Data	Source	Task
Numeracy-600 K [14]	Market comments from Reuters	Magnitude prediction

Table 5.
Number Prediction DataSet.

4. Numeracy embeddings

Embedding or *distributed representation* of words has become basic building blocks for natural language processing in recent years. It represents each word by high-dimension vectors (typically with 50 dimensions or more) of real values. These vectors reflect the meaning of words, such as words with similar meaning are represented by similar[3] vectors. Some researchers have investigated how numeracy itself is modeled in such pre-trained word embedding vectors.

4.1 Task definition

Embedding vectors are also assigned to numerals such as "three", "100", "million", etc. Popular word embeddings like word2vec do not distinguish these numerals from other words, i.e., the learning algorithms for these vectors treat numerals and other words equally. So, it is not obvious these word vectors appropriately reflect the meaning of numbers, such as "100 is larger than 3" and "4 is the next number of 3", etc. *Numeracy embedding* is a task to embed such numerals in appropriate vector representation.

4.2 Investigating pre-trained word vectors

Nowadays, word embeddings learned using huge size of corpus are provided by various researchers. Some researchers investigated how or whether these pretrained word vectors appropriately represent numerals.

Naik et al. [18] used GloVe, FastText, and SkipGram vectors. They compares similarity of embedding vectors for numbers. They used two types of tasks: one is for magnitude, e.g., vector for 4 should be more similar to 3 than 1000000, and the other is for numeration, e.g., vector for *three* should be more similar to 3 than *billion*. Contextualized word vectors were also considered. Wallace et al. [19] found that the pretrained language models for DROP, which is numeracy entailment task mentioned later sections, already captures numeracy, by testing if BiLSTM model with pretrained embedding pass some tests such as list maximum, decoding (e.g., convert the string "five" to 5), taking a sum of two numbers.

4.3 Obtaining word vectors for numerals

On the other hand, developing algorithms specialized to obtain word vectors for numerals beyond pre-trained word vectors have been proposed by some researches in recent years.

Jiang et al. [20] proposed to obtain embedding for numbers by directly applying Skip-Gram models to obtain embeddings for numbers taking into consideration of meaning of numbers by taking weighted average of embeddings which is

---

[3] Similarity of vectors is typically defined by inner product or cosine similarity of vectors.

System	Vector	Algorithm	Task
Naik et al. [18]	Glove, FastText,	Cosine Similarity	Magnitude[a] prediction,
	SkipGram		Numeration[b] prediction
Wallace et al. [19]	NAQANET[c]	biLSTM	List maximum, Decoding, Sum
Jiang et al. [20]	Learned	Cosine Similarity	Magnitude prediction,
			Numeration prediction
			Numeral prediction
Sundararaman et al. [21]	Learned	Cosine Similarity,	Magnitude prediction,
		BiLSTM	Numeration prediction
			List maximum, Decoding, Sum

[a]*e.g., 4 is more similar to 3 than 1000000.*
[b]*e.g., 'three' is more similar to 3 than 'billion'.*
[c]*A system proposed for DROP dataset.*

Table 6.
Numeracy Embedding Systems.

numerically similar to the target number. They find "prototype numbers" by clustering, and represent numbers as a weighted average of these prototypes. Sundararaman et al. [21] proposed to learn embeddings for numbers, which reflect the distance of two numbers in the number line, independently from words.

Table 6 summarizes these previous methods.

5. Numerical reading comprehension and numerical textural entailment detection

More complex tasks such as textual entailment detection or reading comprehension also require treatment of numbers appropriately to answer some of the questions. We first mention on some early works for these tasks and then introduce some recent systems.

5.1 Task definition

Textual entailment detection is a task to find, given some texts, sentences which are true if the given text (called *hypothesis*) are true. The situation become more complicated if the sentences contains numerals because it requires numerical knowledge to understand the meaning of sentence. For example, we can say that the sentence "five people are in the house." is true given the hypothesis "two men and three women are in the house.", but it requires mathematical knowledge that two plus three equals five.

Some early works for this task include numeracy modules. The system by Tsuboi et al. [22] for textual entailment recognition task (RITE) in NTCIR-9 consider temporal expression matching such as "the first half of Nth century" to the appropriate interval. The system by Iftene and Moruz [23] implemented the special rules for numbers which create intervals considering expressions like "more than" or "over" for the Recognizing Textual Entailment (RTE-6) task.

Reading comprehension is a more complicated task, where the system is required to answer various types of questions.

5.2 Numeracy-focused data sets

Aforementioned studies mainly focused on the "range" of the numbers, i.e., they simply treat numbers as points or distributions defined on the number line. However, reading comprehension tasks require more advanced numeric skills such as addition, average, maximum, etc., into language models.

This line of research typically constructs the dataset for numeracy understanding task by selecting numeracy-related data from existing datasets for reading comprehension, natural language inference, or entailment. The selected data contain many questions that require understanding and calculation on numbers beyond simple range- or distribution-based treatment of numbers.

Roy et al. [24] proposed the task of Quantity Entailment, which require numeric reasoning to answer. Their dataset included the corpus from datasets for Recognizing Textual Entailment (RTE) task. They also proposed a method to solve these problems with CRF-based recognition of quantity part of the text, and rule-based recognition of entailment.

Ravichander et al. [25] proposed the EQUATE framework for quantitative reasoning in textual entailment, such as determining "5855 of lambs are back" is correct given the premise "6048 lambs is either black or white and there are 193 white ones." DROP proposed by Dua et al. [26] require systems to do operations such as addition, counting, or sorting. The type of questions and answers in DROP dataset varies widely, such as the question "Where did Charles travel to first" given passages "In 1517, the King sailed to Castle. ... In 1518, he traveled to Barcelona." State-of-the-art methods for reading comprehension performed poorly for these datasets (both of EQUATE and DROP) and the authors concluded that more advanced methods are required for these new tasks for numeric reading comprehension.

Table 7 summarizes these datasets.

5.3 Methods

Given these datasets, more advanced models for them have been proposed. Typical approaches given the recent advance of deep neural network technologies is to use sequence-to-sequence (seq2seq) model for this task. In seq2seq models, the sequence of words can be feed as input directly to the system, then the system also returns another sequence of words as the output. Especially, recent pretrained language models including BERT already contains language models trained on huge amount of text documents, and they can be trained to return appropriate word sequence by being trained on relatively small set of training samples (i.e., the pair of input documents and "correct" or appropriate output for each input.) of a given task.

Rozen et al. [27] reported that performance for existing natural language inference (NLI) datasets can be improved by augmenting the dataset with synthetic adversarial datasets including the ones generated by rule-based replacement of numeric expressions found in the dataset. Geva et al. [28] reported that adding

Data Set	Source	Task
DROP [26]	Wikipedia	Question answering
EQUATE [25]	News, Reddit, Synthetic	Entailment

Table 7.
Numeracy-Focused Data Sets.

System	Method	Base Model
Rozen et al. [27]	Data augmentation	BERT
Geva et al. [28]	Subtask addition	BERT
Ran et al. [29]	Adding reasoning module	NAQANet

Table 8.
Numeracal Reading Comprehension Systems.

synthetic numerical tasks to BERT pretraining steps with fine tuning on DROP dramatically improved the score for DROP. Ran et al. [29] proposed to inject graph-based numerical reasoning module between embedding and prediction modules, which outperformed existing machine reading comprehension models on the DROP dataset.

Table 8 summarizes these approaches.

6. Solving math word problems

Math word problem texts are a typical type of documents that contain numerals and words extensively and require deep understanding of the meaning of numerals. Developing a system that automatically solve math word problems is thus one major research task in this area.

6.1 Task definition

In this task, the problem is given in a text that contains numerals, e.g., "How much How much would it cost to buy 12 apples at 1.1 dollars each?", and systems are required to provide a solution for the problem, e.g., $12 \times 1.1 = 13.2$ dollars. Recent approaches for this task typically use deep neural networks that take a sequence of words as inputs. These inputs are transformed through several layers and used to produce the final output. Variety of output forms are considered by previous methods, including simple seq2seq models (i.e., outputs are also sequences of words) and sequence-to-tree models (i.e., outputs are in tree forms that represent equations to calculate the answers.)

Sequence-to-sequence (seq2seq) is a typical approach for this task. Ling et al. [30] provided their original dataset with 100,000 samples, and proposed a method to generate *answer rationales* which are human-readable instructions to derive the answers using a sequence-to-sequence (seq2seq) model. Saxton et al. [31] investigate the ability of existing sequence to sequence architectures including Transformer for mathematical reasoning (e.g., "Solve 41 + 132") with free-form texts.

Some researchers have tried to produce graphs that represent the mathematical operations to directly produce the answers to the questions. Amini et al. [32] provided a dataset for math word problems called MathQA. They also proposed the sequence-to-Program model to solve this task. The approach by Zhang et al. [33] uses a new architecture called Graph2Tree, which uses graphs constructed from texts independently from BiLSTM encoders. They tested their system on MAWPS data set. [34] Lample and Charton [35] showed that neural models can solve mathematical problems such as symbolic integration and solving differential equations using sequence-to-sequence approaches.

Table 9 summarizes the existing data sets and **Table 10** summarizes the systems for this task proposed so far.

System	Method	Input	Output
Ling et al. [30]	Seq2seq	GRE/GMAT-style	Answer rationales[a]
Saxton et al. [31]	Seq2seq	Free-form text	Solution
Amini et al. [32]	Sequence-to-Program	GRE/GMAT-style	Solution
Zhang et al. [33]	Seq2seq, Graph2Tree	Free-form text	Solution
Lample and Charton [35]	Seq2seq	Equations	Solution

[a]*linguistic instructions to calculate answers.*

Table 9.
Systems for Math Word Problem Solving.

Data	Source
MAWPS [34]	Collected through the Web interface
AQuA [30]	Originally created
MathQA [32]	Enhancing AQuA

Table 10.
Math Word Problem Datasets.

7. Other tasks

Yoshida et al. [36] considered a problem of estimating appropriate units for the numbers found in Wikipedia tables when units were omitted. Elazar and Goldberg [37] considered the problem to infer the omitted head related to numerals such as "It is worth about two million __."

Chen et al. [38] proposed the numeral attachment task, which determine what entity is the number presented in text related. They also proposed the task of numeral categorization, which is to classify numerals presented in financial text into 7 or 17 categories [39].

The task proposed by Chaganty and Liang [40] was to describe given numerals by examples, such as "$131 million is about the cost to employ everyone in Texas over a lunch period."

8. Conclusions

The relations between numerals and words found in text data has been paid little attention compared to other areas in natural language processing. This paper provided the overview of this field ranging from the systems for traditional tasks such as information retrieval tor the relatively recent tasks like reading comprehension.

We categorized the previous researches into 6 types: traditional tasks, numerical common sense acquisition, numeracy embeddings, numerical reading comprehension, solving math word problems, and others. The first two tasks have been studied relatively long time, while the remaining topics is emerging with recent advances of neural language models.

In Section 2, we introduced some previous systems that have numerical modules for traditional tasks like QA, IE, and IR. In Section 3, we introduced numerical common sense acquisition where typical approaches are pattern-based extraction and parameter estimation for language models. In Section 4, numeracy embedding,

where the goal is assigning appropriate real-valued vectors to numerals, was introduced, Section 5 introduced numerical reading comprehension and numerical entailment, that require more advanced numerical understanding of text. The task of solving math word problems, which are typical type of texts that contain numerals extensively, was introduced in Section 6, and Section 7 touched on other unique tasks.

Recent increase of the dataset and resources focusing on numeracy will accelerate the development of the systems with the ability of understanding numeracy in text.

Acknowledgements

Author details

Minoru Yoshida* and Kenji Kita
Tokushima University, Tokushima, Japan

*Address all correspondence to: mino@is.tokushima-u.ac.jp

IntechOpen

References

[1] John M. Prager, Jennifer Chu-Carroll, Krzysztof Czuba, Christopher A. Welty, Abraham Ittycheriah, Ruchi Mahindru: IBM's PIQUANT in TREC2003. TREC 2003: 283-292

[2] Numerical data integration for cooperative question-answering Véronique Moriceau Proceedings of the Workshop KRAQ'06: Knowledge and Reasoning for Language

[3] Somnath Banerjee, Soumen Chakrabarti, Ganesh Ramakrishnan: Learning to rank for quantity consensus queries. SIGIR 2009: 243-250

[4] Sunita Sarawagi, Soumen Chakrabarti: Open-domain quantity queries on web tables: annotation, response, and consensus models. KDD 2014: 711-720

[5] Minoru Yoshida, Issei Sato, Hiroshi Nakagawa, Akira Terada: Mining Numbers in Text Using Suffix Arrays and Clustering Based on Dirichlet Process Mixture Models. PAKDD (2) 2010: 230-237

[6] Anton Bakalov, Ariel Fuxman, Partha Pratim Talukdar, Soumen Chakrabarti: SCAD: collective discovery of attribute values. WWW 2011: 447-456

[7] Eiji Aramaki, Takeshi Imai, Kengo Miyo, Kazuhiko Ohe: UTH: SVM-based Semantic Relation Classification using Physical Sizes. SemEval@ACL 2007: 464-467

[8] Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, Ali Farhadi: Are Elephants Bigger than Butterflies? Reasoning about Sizes of Objects. AAAI 2016: 3449-3456

[9] Dmitry Davidov, Ari Rappoport: Extraction and Approximation of Numerical Attributes from the Web. ACL 2010: 1308-1317

[10] Hiroya Takamura, Jun'ichi Tsujii: Estimating Numerical Attributes by Bringing Together Fragmentary Clues. HLT-NAACL 2015: 1305-1310

[11] Katsuma Narisawa, Yotaro Watanabe, Junta Mizuno, Naoaki Okazaki, Kentaro Inui: Is a 204 cm Man Tall or Small ? Acquisition of Numerical Common Sense from the Web. ACL (1) 2013: 382-391

[12] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, Dan Roth: How Large Are Lions? Inducing Distributions over Quantitative Attributes. ACL (1) 2019: 3973-3983

[13] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, Dan Roth: Do Language Embeddings capture Scales? EMNLP (Findings) 2020: 4889-4896

[14] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, Hsin-Hsi Chen: Numeracy-600K: Learning Numeracy for Detecting Exaggerated Information in Market Comments. ACL (1) 2019: 6307-6313

[15] Taylor Berg-Kirkpatrick, Daniel Spokoyny: An Empirical Investigation of Contextualized Number Prediction. EMNLP (1) 2020: 4754-4764

[16] Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, Xiang Ren: Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. EMNLP (1) 2020: 6862-6868

[17] Georgios P. Spithourakis, Sebastian Riedel: Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers. ACL (1) 2018: 2104-2115

[18] Aakanksha Naik, Abhilasha Ravichander, Carolyn Penstein Rosé,

Eduard H. Hovy: Exploring Numeracy in Word Embeddings. ACL (1) 2019: 3374-3380

[19] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, Matt Gardner: Do NLP Models Know Numbers? Probing Numeracy in Embeddings. EMNLP/IJCNLP (1) 2019: 5306-5314

[20] Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, Kewei Tu: Learning Numeral Embedding. EMNLP (Findings) 2020: 2586-2599

[21] Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, Lawrence Carin: Methods for Numeracy-Preserving Word Embeddings. EMNLP (1) 2020: 4742-4753

[22] Yuta Tsuboi, Hiroshi Kanayama, Masaki Ohno, Yuya Unno: Syntactic Difference Based Approach for NTCIR-9 RITE Task. NTCIR 2011

[23] Adrian Iftene, Mihai Alex Moruz: UAIC Participation at RTE-6. TAC 2010

[24] Subhro Roy, Tim Vieira, Dan Roth: Reasoning about Quantities in Natural Language. Trans. Assoc. Comput. Linguistics 3: 1-13 (2015)

[25] Abhilasha Ravichander, Aakanksha Naik, Carolyn Penstein Rosé, Eduard H. Hovy: EQUATE: A Benchmark Evaluation Framework for Quantitative Reasoning in Natural Language Inference. CoNLL 2019: 349-361

[26] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, Matt Gardner: DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. NAACL-HLT (1) 2019: 2368-2378

[27] Ohad Rozen, Vered Shwartz, Roee Aharoni, Ido Dagan: Diversify Your

Datasets: Analyzing Generalization via Controlled Variance in Adversarial Datasets. CoNLL 2019: 196-205

[28] Mor Geva, Ankit Gupta, Jonathan Berant: Injecting Numerical Reasoning Skills into Language Models. ACL 2020: 946-958

[29] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, Zhiyuan Liu: NumNet: Machine Reading Comprehension with Numerical Reasoning. EMNLP/IJCNLP (1) 2019: 2474-2484

[30] Wang Ling, Dani Yogatama, Chris Dyer, Phil Blunsom: Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. ACL (1) 2017: 158-167

[31] David Saxton, Edward Grefenstette, Felix Hill, Pushmeet Kohli: Analysing Mathematical Reasoning Abilities of Neural Models. ICLR (Poster) 2019

[32] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, Hannaneh Hajishirzi: MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. NAACL-HLT (1) 2019: 2357-2367

[33] Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, Ee-Peng Lim: Graph-to-Tree Learning for Solving Math Word Problems. ACL 2020: 3928-3937

[34] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, Hannaneh Hajishirzi: MAWPS: A Math Word Problem Repository. HLT-NAACL 2016: 1152-1157

[35] Guillaume Lample, François Charton: Deep Learning For Symbolic Mathematics. ICLR 2020

[36] Minoru Yoshida, Kazuyuki Matsumoto, Kenji Kita: Table Topic

Models for Hidden Unit Estimation.
AIRS 2016: 302-307

[37] Yanai Elazar, Yoav Goldberg:
Where's My Head? Definition, Dataset
and Models for Numeric Fused-Heads
Identification and Resolution. Trans.
Assoc. Comput. Linguistics 7: 519-535
(2019)

[38] Chung-Chi Chen, Hen-Hsen Huang,
Hsin-Hsi Chen: Numeral Attachment
with Auxiliary Tasks. SIGIR 2019: 1161-
1164

[39] Chung-Chi Chen, Hen-Hsen Huang,
Yow-Ting Shiue, Hsin-Hsi Chen:
Numeral Understanding in Financial
Tweets for Fine-Grained Crowd-Based
Forecasting. WI 2018: 136-143

[40] Arun Tejasvi Chaganty, Percy
Liang: How Much is 131 Million Dollars?
Putting Numbers in Perspective with
Compositional Descriptions. ACL (1)
2016