Sentiment-basiertes Argument Retrieval

Christian Staudte, Lucas Lange und Ossama Saker

Universität Leipzig - Fakultät für Mathematik und Informatik
Deutschland

Zusammenfassung. Die vorliegende Arbeit stellt einen Lösungsansatz für *Touché 2020: 1st Shared Task on Argument Retrieval - Task (1)* [`https://touche.webis.de`] vor. Grundlegende Anwendung findet ein Word-Embedding (CBOW) basierend auf T. Mikolov et al., 2013 [9]. Weiterhin wird ein neuer Schritt gewagt und versucht, Sentiment Analyse als Hilfsmittel für ein Ranking einzubinden, um zwischen der Qualität einzelner Argumente zu differenzieren.

Schlüsselwörter: Argument Retrieval · Dual Embedding · Sentiment Analysis.

1 Einleitung

Die vorliegende Arbeit stellt einen Lösungsansatz für *Touché 2020: 1st Shared Task on Argument Retrieval - Task (1)*[1] vor. Das Ziel dieses Tasks lautet, für eine gegebene Query möglichst relevante Argumente aus einem Datensatz von 387.692 Argumenten zu filtern und absteigend der Qualität nach zu sortieren.

Für die Entwicklung eines passenden Retrieval-Modells stützt sich die Arbeit auf eine existierende Architektur, die ein Word-Embedding aus einem neuronalen Netzwerk instrumentalisiert. Zusätzlich wird eine Sentiment-Analyse angewandt, um den emotionalen Gehalt jedes Arguments zu bestimmen. Mit diesen Ansätzen verfolgt die Arbeit zwei Ziele: Zum einen soll ermittelt werden, inwieweit sich das Word-Embedding dazu eignet, passende Argumente zu selektieren, und zum anderen soll der Einfluss von Sentiment-Werten für die Qualität der Argumente ermittelt werden.

Die Entwicklung erfolgt mittels Python, Terrier[2] [11], MongoDB und der Google Cloud Natural Language API[3] zur Bestimmung der Sentiment-Werte. Der Quellcode befindet sich im Git-Repository der Gruppe.[4]

Der nächste Abs. (2) beginnt mit einer Übersicht verwandter Arbeiten. Darauffolgend ist die Ausarbeitung so strukturiert, dass zuerst das Argument Retrieval Modell (Abs. 3) näher beleuchtet wird, bevor die darauf bezogene Evaluation in Abs. 4 erfolgt. Der Abschluss (Abs. 5) bildet eine Diskussion des Verfahrens auf Basis der Evaluation und rundet die Arbeit mit einem Fazit ab.

[1] `https://touche.webis.de`
[2] `http://terrier.org/`
[3] `https://cloud.google.com/natural-language/`
[4] `https://github.com/luckyos-code/ArgU`

2 Related Work

Grundlage dieser Arbeit ist ein Word-Embedding (CBOW) nach [9], das sowohl zur Vektorisierung von Argumenten als auch für die Terme einer Query angewandt wird. Mit diesem Ansatz ist es möglich herauszufinden, ob Kontexte zwischen Queries und Argumenten ähnlich sind. Dies führt bei Queries zu Problemen, für die es nur wenig relevante Argumente im Korpus gibt, denn selbst wenn der Kontext ähnlich ist, ist dies kein Garant für die Relevanz eines Argumentes. Aus diesem Grund stützt sich diese Arbeit auf das *Dual Embedsing Space Model* (DESM) von Mitra et al. [10]. Dieses wendet zusätzlich ein BM25-Modell an, um sowohl Kontext als auch Thema besser einzugrenzen. Da nach [12] BM25 für das Argument Retrieval bezüglich DirichletLM und DPH unterlegen ist, wird hier BM25 durch DPH ersetzt.

Die am nächsten verwandten Arbeiten zur Sentiment Analyse im Bereich des Argument Retrieval vefolgen das Ziel eines reinen Opinion Minings, also der Analyse der Haltung eines Arguments [7]. Dadurch siedelt sich die Verwendung primär dort an, wo nicht auf einem bereits mit Haltungen versehenen Korpus gearbeitet wird, sondern eben dieser generiert werden soll (bspw. [1,8,14,16]). Das Übersetzen der Sentiment Analyse auf das Ranking von Argumenten findet hingegen in noch keiner Veröffentlichung Einsatz und wird deshalb ohne Referenzen in Abs. 3.5 implementiert und mit verschiedenen Experimenten (s. Abs. 4) evaluiert, um einen möglichen Nutzen zu bestimmen.

3 Argument Retrieval Modell

Dieser Abschnitt behandelt das Modell zur Akkumulation relevanter Argumente A für eine Query Q. Dieses besteht aus drei Komponenten, die im Folgenden vorgestellt werden: Die Vorverarbeitung (3.1), das DESM (3.2 & 3.3) und die Sentiment-Analyse (3.4 & 3.5).

3.1 Vorverarbeitung

Viele der 387.692 Argumente weisen Syntax- oder Formatierungsfehler auf, die zu Problemen bei der Erstellung von Retrieval Modellen führen können. Um dem entgegenzuwirken, werden die Argumente in einem Preprocessing vorverarbeitet. Dazu werden vorerst die ersten 200 Argumente aus *args-me.json* manuell analysiert, um verschiedene Fehlerarten zu kategorisieren. Mittels dieser Analyse werden im zweiten Schritt Regeln abgeleitet, die auf alle Argumente angewandt werden. Zu diesen Regeln gehören:

1. Das Entfernen bzw. Ersetzen aller URLs
2. Das Löschen von eckigen Klammern und deren Inhalten
3. Das Entfernen bestimmter Sonderzeichen
4. Steht der gleiche Buchstabe in einem Term mehr als zwei Mal hintereinander, werden alle bis auf ein Repräsentant gelöscht, z. B.: *hellooooo* → *hello*

5. Die Korrektur von Komma- und Punktsetzung
6. Das Löschen von zu kurzen Argumente, s. Exp. 4.1.

Dieser Bereinigungsprozess ist darauf ausgelegt, den Text so wenig wie möglich zu ändern, um einen natürlichen Sprachfluss beizubehalten. Die Begründung dafür liegt in der Weiterverarbeitung der Argumente in Abs. 3.4 (Sentimentanalyse). Dort werden Sentiment-Werte mit Hilfe der Google-API extrahiert. Damit diese bestmögliche Resultate liefert, dürfen Argumente nicht zu grob vorverarbeitet werden. Neben dem ersten Preprocessing für die Sentimentanalyse werden Argumente für das eigene Retrieval Modell weiter bereinigt:

6. Zahlen und URLs werden durch Tags ersetzt, z. B.: *123* → *<NUM>*
7. Satzzeichen werden entfernt
8. Terme werden anhand einer Stoppwortliste aussortiert

Diese weiteren Schritte dienen der Verbesserung der Ergebnisse des DESM (Abs. 3.2). Eine Auffälligkeit des Preprocessings ist das Beibehalten von Groß- und Kleinschreibung. Der Grund liegt in möglicherweise entstehenden Wort-Ambiguitäten, s. „US“ → „us“. Derartige Fehler wirken sich auf die Berechnung von Argument-Embeddings aus.

Ein Nachteil, der durch das Beibehalten von Groß- und Kleinschreibung entsteht ist der, dass manche Begriffe verschiedene Schreibweisen besitzen. Dieses Problem wird für Wörter wie „E-Cigarettes“ deutlich. Neben dieser Schreibweise findet man sowohl „e-cigarettes“, als auch „E-cigarettes“ im Korpus wieder. Da Wörter in den vorgegebenen Queries aus *topics.xml* grundsätzlich mit Großbuchstaben beginnen, werden diese neben dem oben beschriebenen Preprocessing zusätzlich bearbeitet: Unter Verwendung des *Natural Language Toolkits* (NLTK)[5] werden zuerst POS-Tags bestimmt. Gehört ein Wort keiner der Noun-Klassen an, wird der Anfangsbuchstabe klein geschrieben. Zusätzlich werden Wörter dupliziert und vollständig kleingeschrieben, sofern sie mehr als einen Großbuchstaben beinhalten. Damit gilt bspw.: „Vaping E-Cigarettes Safe“ → „vaping e-cigarettes E-Cigarettes safe“.

3.2 Das DESM

Mitra et al. verwenden neben BM25 ein *Continuous Bag of Words* (CBOW) Modell [9]. Mit diesem werden Argumente, die aus beliebig vielen Termen bestehen, auf einen Vektor abgebildet, indem der geometrische Schwerpunkt $\overline{A}$ eines Arguments A berechnet wird:

$$\overline{A} = \frac{1}{|A|} \cdot \sum_{a_j \in A} \frac{a_j}{||a_j||} \tag{1}$$

$|A|$ bezeichnet die Menge der Terme in A und a_j die Embeddings, die in der Formel normalisiert werden. Mit der Normalisierung schränkt man den Einfluss

[5] https://www.nltk.org/

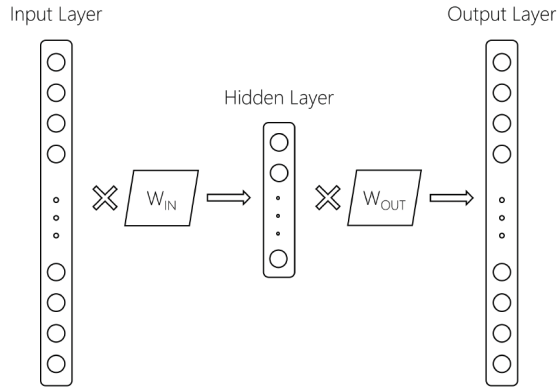

Abb. 1: W_{IN} sind die Standard-Embeddings für Wörter. In DESM werden auch die Vektoren aus W_{OUT} verwendet, um Ergebnisse zu optimieren.

frequenter Terme ein[6]. Im nächsten Schritt werden Kosinus-Ähnlichkeiten zwischen Termen der Query Q und $\overline{A}$ berechnet. Dies ergibt den DESM-Score:

$$DESM(Q, A) = \frac{1}{|Q|} \cdot \sum_{q_i \in Q} \frac{q_i^T \overline{A}}{||q_i^T|| \cdot ||\overline{A}||} \tag{2}$$

Zur Optimierung von Suchanfragen können die Argument-Vektoren $\overline{A}$ vorberechnet werden. Um nun die Modelle BM25 und DESM zu kombinieren, bestimmt man pro Query und Argument zwei Scores, die abhängig von einem Hyperparameter α kombiniert werden:

$$score_{Q,A} = \alpha \cdot DESM(Q, A) + (1 - \alpha) \cdot BM25(Q, A) \tag{3}$$

Mitra et al. merken jedoch selbst an, dass CBOW sich dazu eignet, einen passenden Kontext zu finden und BM25, um Feinheiten zu filtern. Daher verzichtet diese Arbeit auf die parallele Ausführung beider Modelle und wendet stattdessen DESM und DPH sukzessiv an. Damit bestehen die finalen Scores der Argumente nur aus DPH-Werten. Dies führt aber dazu, dass eine Größe N für den DESM-Ergebnisraum E_{DESM} bestimmt werden muss, s. Exp. 4.2.

Um nun zu verhindern, dass E_{DESM} Argumente enthält, die im Ergebnisraum von DPH (E_{DPH}) vollkommen irrelevant sind[7], werden die Ergebnisse des DPH pro Anfrage auf die Top 1.000 eingeschränkt. Damit geht aber ein Problemen einher: Sind E_{DESM} und E_{DPH} disjunkt, können für eine Query auch keine Ergebnisse gefunden werden – dieses Problem wird hier nicht weiter behandelt.

[6] Nach Schakel und Wilson besitzen Terme, die häufig im gleichen Kontext vorkommen, längere Vektoren [13].

[7] Dies tritt dann ein, wenn E_{DESM} den Kontextraum nicht exakt einschränken kann.

3.3 IN und OUT Vektoren

Konträr zu den meisten anderen Ansätzen nutzen Mitra et al. nicht nur W_{IN} als Term-Vektoren (IN), sondern auch W_{OUT} (OUT), s. Abb. 1.

> [...] the IN-IN (or the OUT-OUT) cosine similarities are higher for words that are typically (by type or by function) similar, whereas the IN-OUT cosine similarities are higher for words that co-occur often in the training corpus (topically similar). [10]

Sie selbst nutzen einen $DESIM_{IN-OUT}$ Score. Das bedeutet, Query-Terme werden mit IN-Vektoren kodiert und Argument-Terme mit OUT-Vektoren. Um festzustellen, welcher Ansatz für das Argument Retrieval sinnvoller ist, werden $DESIM_{IN-IN}$ und $DESIM_{IN-OUT}$ in Exp. 4.3 miteinander verglichen. Dabei zeigt sich, dass $DESIM_{IN-OUT}$ zwar neue Argumente liefern kann, im Gesamten jedoch ein wenig schlechter als $DESIM_{IN-IN}$ abschneidet. Folglich wird hier weiterhin $DESIM_{IN-IN}$ genutzt.

3.4 Sentiment Analyse

Die Sentiment Analyse hat das Ziel, den emotionalen Charakter eines Textes herauszufinden und zu bewerten. Wie im Abs. 2 vorgestellt, ist die Verwendung für das Opinion Mining besonders präsent. Dem liegt die Annahme zugrunde, dass Diskussionen und damit auch Argumente, grundlegend emotional aufgeladen sind [4,17]. Versteht man nun die Emotionalität eines Arguments als einen Punkt auf einer Skala, welche die Bereiche negativer, positiver und auch neutraler Emotion abdeckt, so schließt diese Auffassung neben pro und contra auch eine neutrale Haltung ein. Angewandt auf das Ranking von Argumenten versucht diese Arbeit aus den Sentiment-Werten – also aus der Emotionalität gemessen an einer Skala – Rückschlüsse auf die Güte eines Arguments zu ziehen. Die Hypothese lautet, dass eine emotionale Bindung an ein Thema auf eine Involviertheit und dadurch stärkere Argumentation hindeuten kann [2,4,17,18]. Die Ergebnisse sollen letztlich dazu führen, die Sortierung aus DESM und DPH mit Sentiments zu kombinieren und zu verfeinern.

Zur Analyse können eigene Modelle mittels NLTK trainiert werden, was den Vorteil bietet, eigene Trainingsdaten selektieren zu können und darüber die Analyse an das Einsatzgebiet anzupassen. So kann es sinnvoll sein, domänenspezifische Daten wie Filmkritiken zu wählen, um den dortigen Ton der Argumente zu lernen. Für dieses Projekt ist ein solches Modell nachteilig. Die Diskussionen im Datensatz sind thematisch vielfältig, weswegen ein optimales Training auf allgemeinen und ausgewogenen Daten beruhen sollte.

Das Tool SentiStrength[8] wurde bereits in wissenschaftlichen Arbeiten angewandt und auch hier in Erwägung gezogen. Nicht implementiert wurde es letztlich aufgrund der Nachteile im Bereich politischer Texte [15].

[8] `http://sentistrength.wlv.ac.uk/`

Gewählt wurde stattdessen die Sentiment Analyse der Google Cloud Natural Language API, welche für eine breite Palette von Anwendungsfeldern bereitgestellt wird und daher für den vielfältigen Inhalt der Diskussionen im Datensatz geeignet ist. Einzig die Kosten des Service stellen für dieses Projekt ein Problem dar und führen zu Restriktionen. Die API wertet jede 1.000 Zeichen eines Textes als eine Anfrage und berechnet dementsprechend die Kosten an der Anzahl der Anfragen [6]. Unser Budget erlaubt eine Anfrage pro Argument zu stellen, was bedeutet, dass Argumente auf 1.000 Zeichen gekürzt werden müssen. Ausgeführt auf dem bereinigten Datensatz im Umfang von 297.018 werden mehr als die Hälfte, genau 155.216, der Argumente gekürzt, was zu einem partiellen Informationsverlust führt[9]. Nach einem manuellen Vergleich, in welchem gekürzte und ungekürzte Argumente hinsichtlich der Emotionalität analysiert wurden, konnte kein ausschlaggebender Unterschied erkannt werden. Das führt zur Annahme, dass ein Argument nach ≈164 Wörtern eine erkennbare Haltung (Sentiment) angenommen haben sollte.

Die Antwort auf eine Anfrage enthält Sentiment-Werte für jeden einzelnen Satz, sowie einen dazugehörigen Magnitude-Wert, welcher angibt, wie stark die Emotion ist. Ausgehend dieser Teilbewertungen wird der allgemeine Sentiment- und Magnitude-Wert des Dokuments bestimmt. Der Magnitude-Wert ist hier irrelevant, da dieser aus allen Sätzen aufsummiert wird und durch die Kürzung auf 1.000 Zeichen im Vergleich starken Ungenauigkeiten unterliegt. Der Sentiment-Score wiederum gibt die Richtung der Emotion an und kann damit auf positiv, negativ oder neutral (Pos, Neg, N) hindeuten. Er berechnet sich aus dem normalisierten Durchschnitt der Sätze, wobei Werte zwischen -0.1 und 0.1 als N gewertet werden. Werte kleiner -0.1 bzw. größer 0.1 übermitteln stattdessen eine negative bzw. positive Richtung. [5]

Als Ergebnis der Analyse kann jedem Argument ein entsprechendes Sentiment-Ergebnis zugeordnet werden. Die Verteilung der Sentiment-Werte ist in Abb. 2 dargestellt. Die Mehrheit der Argumente befindet sich demnach im neutralen Bereich und die Anzahl nimmt Richtung der Extrema -1 und 1 stark ab. Auffällig ist die Neigung ins Negative, obwohl die Mengen der Pro und Con Argumente im Datensatz nahezu gleich sind[10]. Ob weitere Experimente mit Sentiment-Werten einen Mehrwert bringen können, wird in Exp. 4.4 beantwortet, indem ihr Informationsgehalt bekannten Informationen gegenübergestellt werden.

3.5 Integration von Sentiments in DESM und DPH

Auch wenn die Experimente (4.5, 4.6) darauf hindeuten, dass Sentiments nur sehr eingeschränkt bei der Selektion qualitativ hochwertiger Argumente helfen, werden trotzdem zwei Ansätze getestet und mittels TIRA ausgewertet:

[9] Die durchschnittliche Wortlänge im Englischen beträgt 5,1 Zeichen, was mit einem Leerzeichen zwischen den Wörtern zu einer maximalen Länge von etwa 164 Wörtern für 1.000 Zeichen führt, abzüglich etwaiger Satzzeichen [19]. Zum Vergleich: Der Paragraph mit dieser Fußnote enthält ungefähr 1.100 Zeichen.

[10] Im Datensatz sind 53% Pro und 47% Con Argumente.

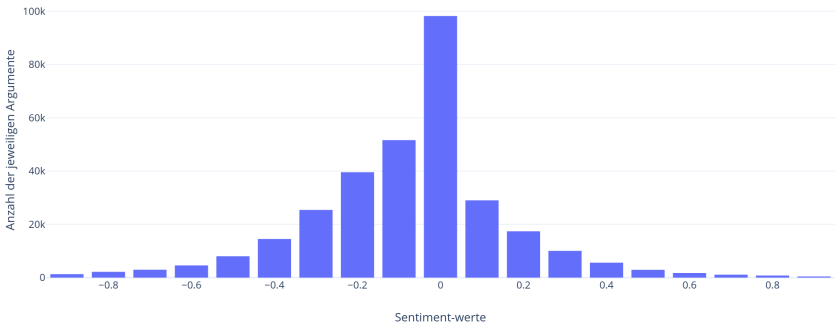

Abb. 2: Verteilung der Sentiment-Werte im bereinigten Datensatz.

1. Sentiments mit hohen Ausschlägen (nahe 1 oder -1) werden besser bewertet.
2. Neutralere Sentiments werden besser bewertet.

Dazu kommt folgende Formel zum Einsatz:

$$DPH^*_{Q,A} = DPH_{Q,A} \pm DPH_{Q,A} \cdot \frac{S_A}{2} \tag{4}$$

S_A bezeichnet den Sentiment Wert für Argument A. Addition findet in Fall (1) statt und Subtraktion in Fall (2). Die Ergebnisse befinden sich im Abs. 4.7 unter Tab. 4.

4 Evaluation

Die Evaluation umfasst in erster Linie 6 Experimente, die Aufschluss über eigene Hypothesen und Ideen liefern sollen. Die ausgewerteten Daten befinden sich in vollem Umfang (Argument IDs, individuelle Scores, ...) auf dem Git-Repository[11]. Am Ende dieses Kapitels folgt die Evaluation des Systems auf TIRA[12].

4.1 Exp. 1: Kurze Argumente

Hypothese: *Kurze Texte stellen keine relevanten Argumente dar und sollten gelöscht werden, um Fehler im Retrieval-Prozess einzuschränken.*
Dazu werden die ersten 200 Argumente, die nach dem Preprocessing maximal 25 Terme aufweisen ($|A_i| \leq 25$) manuell ausgewertet. Von diesen besitzen insgesamt

[11] `https://github.com/luckyos-code/ArgU/doc/experiments`
[12] Tira User *ir-lab-ul-t1-detroitnitz*, `https://www.tira.io`

nur vier Stück einen argumentativen Charakter – ungeachtet der Qualität. Die restlichen Argumente tragen keine neuen Inhalte zur Debatte bei. Insgesamt gilt für 90.674 der 387.692 Argumente: $|A_i| \leq 25$. Entfernt man diese, gehen nach der Schätzung ungefähr 1.800 (möglicherweise) relevante Argumente verloren. Dieser Wert soll hier vernachlässigt werden.

4.2 Exp. 2: Bestimmung von N für E_{DESM}

Hypothese: *Der Ergebnisraum für DESM darf weder zu groß noch zu klein sein. Ist er zu klein, werden relevante Argumente ignoriert. Ist er zu groß, werden womöglich Kontextunabhängige Argumente selektiert.*
In diesem Versuch werden die ersten 5 Topics betrachtet:

Q_1 Teachers Get Tenure
Q_2 Vaping E-Cigarettes Safe
Q_3 Insider Trading Allowed
Q_4 Corporal Punishment Used Schools
Q_5 Social Security Privatized

Für diese wird jeweils ein $DESM_{IN-IN}$ mit verschiedenen N-Werten ausgeführt: $N \in \{100, 500, 1000\}$. Nach dem Retrieval-Prozess werden die Top-20 Argumente mittels NDCG@20 ($r : A \times Q \to \{0, 1, 2, 3\}$)[13] und Precision@20 ausgewertet. Findet das Modell weniger als 20 Argumente, wird der Wert dementsprechend angepasst. Tab. 1 zeigt den Einfluss von N auf den geg. Topics.

Für Q_2, Q_4 und Q_5 steigt die Precision mit steigendem N. Q_1 erfährt in dieser Hinsicht leichte Schwankungen, während Q_3 keine brauchbaren Ergebnisse liefern kann. Der NDCG steigt für Q_1 und Q_2 an, wobei Q_2 bei $N = 500$ einen besonders hohen Ausschlag liefert. Für Q_4 und Q_5 bleibt dieser Wert nahezu konstant. Die Precision, die mit steigendem N tendenziell zunimmt (insbesondere für Q_4 und Q_5) ist Anlass dafür, das System mit einem Kontext der Größe $N = 1000$ auszuführen.

4.3 Exp. 3: IN- und OUT-Embeddings

Hypothese: *Ein IN-OUT Embedding kann den Kontext für Queries besser einfangen und ermöglicht bessere Messwerte als IN-IN.*
Nachdem die Größe des Ergebnisraumes in Experiment 4.2 festgelegt wurde, werden die Ergebnisse für $N = 1000$ mit denen eines IN-OUT-Embeddings bei gleichem N-Wert verglichen. Der IN-IN Ergebnisraum zeigt im oberen Teil der Tab. 1 Probleme für Q_3 auf. Dies liegt daran, dass nicht der korrekte Kontextbereich zu *Insider Trading* gefunden wird, sondern der zu *free trading* und *carbon emission trading* (s. Argument-IDs im Git). Demzufolge sind bisher nahezu alle gefundenen Argumente für dieses Topic irrelevant.

[13] 0) Keine Relevanz; 1) Schneidet das Thema an; 2) Trifft das Thema, geht aber auch auf andere Dinge ein; 3) Passt perfekt zum Thema und geht gezielt auf die Debatte ein. – Diese Auswertung beinhaltet noch keine Analyse über die inhaltliche Qualität.

Tabelle 1: Precision@20 und NDCG@20 für die ersten 5 Queries aus *topics.xml*. $N = 1000^*$ sind die Ergebnisse für ein IN-OUT-Embedding. Auffällig niedrig sind die Ergebnisse für Q_3 bezüglich aller Werte für N. Der Fall, dass $Prec = 0$ ist und $NDCG > 0$ wird möglich, da Argumente mit $r = 1$ keinen positiven Einfluss auf die Precision, aber auf NDCG besitzen. Die Precision umfasst nur Argumente, für die $r > 1$ gilt. Eine nähere Auswertung der Ergebnisse für Q_3 folgt in Exp. 4.3.

N	Q_1		Q_2		Q_3		Q_4		Q_5	
	Prec	NDCG	Prec	NDCG	Prec	NDCG	Prec	NDCG	Prec	NDCG
100	0,6	0,85	0,25	0,77	0,0	1,0	0,4	0,96	0,6	0,92
500	0,35	0,87	0,2	0,88	0,0	0,63	0,4	0,95	0,8	0,92
1000	0,45	0,94	0,35	0,82	0,0	0,36	0,9	0,97	0,9	0,92
1000*	0,30	0,99	0,40	0,78	0,0	-	0,9	0,97	0,75	0,93

Ein Ziel des IN-OUT Embeddings soll es sein, die Qualität des Retrievals für Q_3 zu stärken und die Qualität für andere Topics zumindest beizubehalten. Als Messwerte für die Relevanz dienen auch hier Precision@20 und NDCG@20, um eine Vergleichbbarkeit zu gewährleisten. Die Ergebnisse dazu sind auch in Tab. 1 unter $N = 1000^*$ abgebildet. Wie man sehen kann, verschlechtert der neue Kontext die Ergebnisse, statt sie zu verbessern. Nicht nur wird die Precision für Q_1 und Q_5 schlechter, auch besitzen alle gefundenen Argumente für Q_3 eine Relevanz von 0. Aus diesem Grund wird das Modell weiter mit $DESM_{IN-IN}$ ausgeführt.

4.4 Exp. 4: Mehrwert durch Sentiments

Hypothese: *Sentiment-Werte bieten einen Mehrwert im Vergleich zu bereits vorhanden Informationen.*
Dafür werden die Werte mit der im Datensatz gegebenen Haltungen (Pro oder Con) verglichen. Tab. 4 zeigt die Ergebnisse des Experiments. Direkt zu Anfang lässt sich eine Erweiterung um die Bewertungsebene N (Neutral) feststellen, welche nicht direkt mit Pro oder Con assoziert werden kann, aber in der Analyse mit 57% den Großteil der Argumente umfasst und für Pro und Con ausgeglichen auftritt. Bei *Pos* und *Neg* ist feststellbar, dass nur etwas mehr als die Hälfte der Argumente ihre verwandte Haltung (Pro, Con) emotional widerspiegeln. Erneut sieht man, dass fast dreimal mehr Argumente Neg (32%) bewertet werden als Pos (11%). Eine Vermutung ist, dass sich Con Argumente häufiger ausschlaggebend von N unterscheiden, während Pro Argumente sich inhaltlich dem Neutralen annähern.

Aus dem Experiment geht hervor, dass durch die Unterschiede zum Datensatz ein Mehrwert an Informationen entsteht. Die neue Kategorie der neutralen Argumente N, sowie die abweichende Einteilung der Pro/Con Argumente in Pos/Neg, bieten eine neue Grundlage. Erweitert wird diese in Form der Sentiment-Werte selbst, welche nicht nur auf Pos oder Neg abbilden, sondern als Skala fungieren.

Tabelle 2: Vergleich der Haltungen (Pro, Con) von Argumenten mit ihren Sentiment-Werten (Pos, Neg, N) nach Abs. 3.4.

	Anteil	Haltung		Verhältnis
Sentiment	Datensatz	Pro	Con	Pro-Con
N	57%	89.841	79.311	53%-47%
Pos	11%	19.465	13.941	58%-42%
Neg	32%	47.548	46.912	50%-50%

4.5 Exp. 5: Der Einfluss von Sentiment-Werten

Hypothese: *Argumente, die einen großen Sentiment-Wert aufweisen sind von minderer (1) bzw. höherer Relevanz (2).*
Um den Einfluss der Sentiments zu bewerten, werden die Ergebnisse aus Tab. 1 weiter verwendet. Um jedoch eine Verfälschung der folgenden Ergebnisse durch ein unzureichendes Retrieval (s. Q_3) zu vermeiden, werden nur die Top-10 relevantesten Argumente absteigend aus Q_1, Q_4 und Q_5 verwendet. Für die relevantesten Argumente muss gelten, dass $r \geq 2$. Demzufolge können aus Q_1 nur 8 Argumente genutzt werden.

Nach der Selektion wird die inhaltliche Qualität jedes Arguments manuell mit $(r^* : A \times Q \to \{0, 1, 2, 3\})$ evaluiert[14] und anschließend mit den Sentiment-Werten abgeglichen. In Tab. 3 werden die Relevanzwerte mit der jeweiligen Menge gefundener Sentiment-Werte aufgeführt. Die Auswertung beinhaltet zwar nur 28 Argumente, trotzdem kann man sehen, dass die Werteverteilungen für Sentiments und Magnitudes unabhängig von r^* (mit wenigen Ausnahmen) ähnlich sind. Man kann maximal annehmen, dass leicht emotionale (negative) Sentiment-Werte häufiger auf relevante Argumente hindeuten (s. Exp. 4.7). Allgemein sollte man jedoch davon ausgehen, dass Sentiments alleine kein eindeutiger Indikator für die Gesamtqualität eines Arguments sind. Das heißt, ein emotional geschriebenes Argument kann trotzdem qualitativ relevant sein, indem es fundierte Prämissen auflistet, s. Argument *e7b98175-2019-04-18T14:36:18Z-00002-000* ($sent = -0,5;\ r^* = 3$) und neutral formulierte Argumente können gleichzeitig qualitativ schlechter sein, indem sie zum Beispiel auf dubiosen Prämissen basieren. Ein zusätzliches Experiment soll feststellen, ob besonders große Ausschläge der Sentiments ($|sent| > 0,5$) ein besserer Indikator für die Qualität sein kann.

[14] Zur Bemessung der Qualität von Argumenten dienen die Definitionen nach Richard Epstein, s. http://faculty.uncfsu.edu/jyoung/what_is_a_good_aesrgument.htm. Argumente mit $r^* = 3$ benötigen (möglichst wissenschaftlich) fundierte Prämissen. Ein Argument mit $r^* = 0$ ist entweder ein zirkulierendes Argument oder eines, dessen Prämissen nicht vertrauenswürdig / dubios ist. Neben diesen Merkmalen ist die sprachliche Qualität wichtig (Rhetorik und Dialektik). Diese nimmt mit absteigendem r^* ab.

Tabelle 3: Sentimentanalyse mit Sentiments und Magnitudes für die Top-10 relevantesten Argumente von Q_1, Q_2 und Q_3.

Panel A: Ergebnisse als Wertmenge für jeden Relevanzwert r^*. $r^* = 0$ ist nicht vertreten, da keines der betrachteten Argumente dieser Kategorie entspricht.

r^*	Sentiments	Magnitudes
1	0,0 0,0 0.3 0,1 -0,2 0,0 0,0 0,0 0,0 0,0 0,0 -0,1 -0,2	2,8 1,8 2,3 3,0 11,0 27,0 4,8 4,8 3,8 0,0 1,5 19,7 14,8
2	-0,2 0,0 -0,2 0,0 0,3 0,0 0,0 -0,1	2,4 3,0 4,1 2,6 1,4 3,8 5,0 4,0
3	-0,1 -0,5 -0,2 -0,2 0,0 -0,1 -0,1	4,6 5,7 2,6 3,3 1,8 3.7 3,5

Panel B: Statistische Auswertung der Ergebnisse aus Panel A.

r^*	Sentiments			Magnitudes		
	∅	Median	Varianz	∅	Median	Varianz
1	-0,008	0,0	0,016	7,485	3,8	68,085
2	-0,025	0,0	0,025	3,288	3,4	1,324
3	-0,171	-0,1	0,026	3,6	3,5	1,627

4.6 Exp. 6: Der Einfluss extremer Sentiments

Im letzten Experiment wird analysiert, ob Sentiments mit $|sent| > 0,5$ passende Indikatoren für die Qualität von Argumenten sein können. Dazu werden 100 Argumente selektiert und bezüglich der dazugehörigen Debatte manuell ausgewertet.

Neben dem Faktor, dass nur wenige Argumente ein derart starkes Sentiment aufweisen fällt auf, dass ernste Diskussionen mit steigendem Sentiment abnehmen, während jene zum reinen Vergnügung deutlicher herausstechen (bspw. „Rap Battle“). Außerdem ist ein Trend erkennbar, dass stark negative Argumente öfters *Trolle* und abfällige Kommentare enthalten. Bei positiven Argumenten läuft es meist auf simples anpreisen von Diskussionsgegenständen und sogar Unbeteiligtem hinaus. Qualitativ hohe Argumente sind über den gesamten Bereich verteilt und lassen sich nicht einem höheren oder niedrigeren Sentiment-Wert zuordnen. Für die kommenden zwei Argumente gilt jeweils: $sent = -0.7$. Sie sollen beispielhaft die Unterschiede in der Art der Argumentation darstellen, trotz gleichem Sentiment:

1. *a039e5a5-2019-04-18T19:00:50Z-00005-000*
   The salad could still be unhealthy and have too much dressing. Not „that“ bad implies that it is bad, just not terrible. Somewhat bad is still bad. The resolution is negated.

Tabelle 4: Ergebnisse der Tira-Evaluation für die Implementierung der Runs R1, R2 und R3.

	R_1	R_2	R_3
nDCG	0.235	**0.254**	0.231
nDCG@5	0.063	**0.116**	0.022
nDCG@10	0.075	**0.152**	0.065
QrelCoverage@10	0.347	**0.673**	0.367

2. *2428564f-2019-04-18T19:40:12Z-00001-000*
   n1 your an ediot n2 your fat n3 your muclly and look ugly n4 i coulnt be bothered reading your long argument because i rather go play a video game than listen to u \

Klar festzustellen ist, dass die Güte von Argumenten nicht generell an der Höhe ihres Sentiments bemessen werden kann, wie auch schon in Exp. 4.5 gezeigt. Eine Implementierung zum Ranking hebt ggf. genauso die schlechten, wie guten Argumente an. Doch der tatsächliche Effekt einer Implementierung wird im Folgenden evaluiert.

4.7 Evaluation von TIRA

In diesem Abschnitt folgt die finale Evaluation. Diese soll nicht nur Auskunft darüber geben, wie gut das angepasste DESM-Modell abschneidet, sondern auch den Einfluss der Sentiments abschließend darstellen. Dazu werden drei Implementierungen mit ihren Runs analysiert:

R_1 Ohne Sentiments
R_2 Emotional ist besser (Abs. 3.5 Fall 1)
R_3 Neutral ist besser (Abs. 3.5 Fall 2)

Die Ergebnisse[15] zu diesen Runs befinden sich in Tab. 4. R_3 liefert für fast alle Messwerte das schlechteste Ergebnis, bis auf $QrelCoverage$@10 im Vergleich zu R_1. Überraschenderweise ist R_2 bezüglich R_1 in allen Punkten besser. Das heißt, Argumente mit einer gewissen Emotionalität können zur Verbesserung der Retrieval-Qualität beitragen.

5 Diskussion und Fazit

Trotz manueller Evaluationen, die auf die Eingeschränktheit der Sentiment-Analyse hinweist[16], schließt Abs. 4.7 mit der Erkenntnis ab, dass emotionale

[15] Durch die zufällige Initialisierung des CBOW-Modells gibt es bei mehrfacher Ausführung minimale Schwankungen in den Ergebnissen.

[16] Sentiments scheinen stark durch einen Bias für bestimmte Key-Words geprägt zu sein.

Argumente qualitativ hochwertiger sein können und somit besser gerankt werden sollten. Neutrale Argumente waren im Schnitt weniger relevant. Dieses Ergebnis entspricht den in [2,4,17,18] formulierten Annahmen, dass eine emotionale Bindung an ein Thema auf eine Involviertheit und dadurch stärkere Argumentation hindeuten kann. Trotzdem gilt, dass Sentiments alleine kein eindeutiger Indikator für die Qualität eines Arguments sind (Abs. 4.5 & Abs. 4.6).

Das gezeigte Modell kann in zukünftigen Arbeiten in vielerlei Hinsicht angepasst werden: Zum einen kann man den Parameter N zur Festlegung der Größe von E_{DESM} optimieren. Auch kann man aktuellere Embedding-Modelle wie FastText [3] anwenden, um Queries und Argumente zu kodieren. Um das finale Ranking zu verbessern, kann man den Einfluss der Sentiment-Werte anhand Formel 1 anpassen. Außerdem kann man zusätzliche Features wie die Qualität und Diversität im Argument referenzierter Quellen integrieren. Auch die Magnitude der Sentiment-Auswertung kann eine Rolle spielen, sofern man die gesamten Argumente ohne Kürzungen auswerten kann.

Literatur

1. Anand, P., Walker, M., Abbott, R., Tree, J.E.F., Bowmani, R., Minor, M.: Cats rule and dogs drool!: Classifying stance in online debate. In: Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis. pp. 1–9. Association for Computational Linguistics (2011)
2. Blanchette, I., Caparos, S.: When emotions improve reasoning: The possible roles of relevance and utility. Thinking & Reasoning **19**(3-4), 399–413 (2013)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
4. Gilbert, M.A.: Coalescent argumentation. Routledge (2013)
5. Google Ireland Limited: Natural Language API Basics. `https://cloud.google.com/natural-language/docs/basics#sentiment_analysis` (2019), [Online; accessed 26-02-2020]
6. Google Ireland Limited: Pricing. `https://cloud.google.com/natural-language/pricing` (2020), [Online; accessed 26-02-2020]
7. Lawrence, J., Reed, C.: Argument mining: A survey. Computational Linguistics **45**(4), 765–818 (2020)
8. Liao, X., Cao, D., Tan, S., Liu, Y., Ding, G., Cheng, X.: Combining language model with sentiment analysis for opinion retrieval of blog-post. In: TREC. pp. 211–213 (2006)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013), `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`
10. Mitra, B., Nalisnick, E., Craswell, N., Caruana, R.: A dual embedding space model for document ranking (2016)
11. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Losada, D.E., Fernández-Luna, J.M. (eds.) Advances in Information Retrieval. pp. 517–519. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
12. Potthast, M., Gienapp, L., Euchner, F., Heilenkötter, N., Weidmann, N., Wachsmuth, H., Stein, B., Hagen, M.: Argument search: Assessing argument relevance. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1117–1120. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331327, `https://doi.org/10.1145/3331184.3331327`
13. Schakel, A.M.J., Wilson, B.J.: Measuring word significance using distributed representations of words (2015)
14. Somasundaran, S., Wiebe, J.: Recognizing stances in ideological on-line debates. In: Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text. pp. 116–124. Association for Computational Linguistics (2010)
15. Thelwall, M.: The heart and soul of the web? sentiment strength detection in the social web with sentistrength. In: Cyberemotions, pp. 119–134. Springer (2017)

16. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. Journal of the American Society for Information Science and Technology **62**(2), 406–418 (2011)
17. Villata, S., Cabrio, E., Jraidi, I., Benlamine, S., Chaouachi, M., Frasson, C., Gandon, F.: Emotions and personality traits in argumentation: an empirical evaluation 1. Argument & Computation **8**(1), 61–87 (2017)
18. Walton, D.: The place of emotion in argument. Penn State Press (2010)
19. Wolfram Alpha: average word length in the English language. `https://www.wolframalpha.com/input/?i=average+word+length+in+the+English+language` (2020), [Online; accessed 26-02-2020]