* **Großes Interesse an:**
  + Knowledge Graphs und wie diese für Machine Learning/Deep Learning genutzt werden können
  + Wie arbeiten die Algorithmen mit dem Tuples? 🡪Embedding🡪bis jetzt nur mit Features und Datensätzen in Tabellarischer Form gearbeitet
  + CNN (Convulotional Neural Network) for Knowledge Graphs
  + Wie lassen sich ML-Algorithmen nachvollziehen?/wie kommen die Ergebnisse zustande? 🡪 werde ich dafür auch ML algorithmen implementieren
  + Weitere Datenquellen in eine Knowledge Graph einbinden🡪 wie werden Connections am besten erstellt
    - Kann dies vielleicht auch automatisiert werden?
  + Kombination von verschiedenen Knowledge Graphs
  + Von relationalen Datenbanksystemen zu Knowledge Graphs
  + Erstellung von Knowledge Graphs aus bereits existierenden Datenquellen in der Industrie
* **Nicht so großes Interesse an:**
  + Laufzeitanalysen (🡪Komplexitätsanalyse)
  + Mathematischen Gleichungen, zu viel Mathe
  + Embedding ist sehr Theorie-lastig
* **Allgemeine Fragen:**
  + Was würde ich in den einzelnen Unterthemen programmieren ?
  + Werden auch mit Cloud-Systemen gearbeitet?
  + Werden Cloud-Services genutzt oder eigene Server? (🡪AWS Cloud Services)
  + Welche Frameworks werden sonst noch so genutzt ? (🡪 Hadoop, Spark?)
* **Subtopic 1:**
  + DL-Learner aus Subtopic 1 wurde in Java geschrieben 🡪 muss ich in diesem Subtopic auch in Java schreiben?
  + Auch selber Cipher Queries erstellen?
  + Wie gut lassen sich derzeit beispielsweise Ergebnisse von Neuronalen Netzen bereits erklären?
  + Warum wurde eine Prediction1 gemacht anstelle von einer anderen Prediction2
  + Welche Faktoren sollten geändert werden, um meine nächste Prediction besser zu machen?
  + Wie tragen die Merkmale zum Ergebnis bei und wie interagieren sie miteinander?
  + Werden in diesem Subtopic auch ML algorithmen geschrieben?
  + Ist dieses Subtopic für Projekt DAIKIRI?
* **Subtopic 2:**
  + Befasst sich nur mit den KG selbst, aber sehr tiefgreifend, oder?
  + sehr mathematisch dieses Thema?
* **Subtopic 3:**
  + In welchem Projekt findet sich diese Forschungsarbeit wieder?
  + Was könnte ich hier programmieren?
  + Welche Art von Algorithmen sollen auf dem CauseNet angewendet werden?

# Future work

## CauseNet

* Future work **may use the graph for** 
  + **causal reasoning**
  + **computational argumentation**
  + **multi-hop question answering, and more.**
* we found that known unknowns in texts are often expressed in terms of adjectives
  + leave it to future work **to extract known unknowns from texts**, too.
* Future work could utilize our causality **graph to analyze not only the sentiment of causal relations or causal paths but also the influence of causality on sentiment in sentences**
* While we have exemplified its benefit for causal question answering, future work **may utilize it in the context of explainable AI, automated decision making, and much more**.
* Neben den explizit ausgedrückten Kausalitäten auch andere mit einbeziehen:
  + (1) implicitly rather than explicitly
  + (2) across sentences
  + (3) differently across domains, or
  + (4) using negation
* model causality in terms of entities instead of the form of noun phrases, as is done by Wikidata and Dbpedia
* We include detailed provenance information for all causal relations
  + This includes claims that HIV is caused by homosexuality or that autism is caused by vaccination.
* Eigene Ideen:
  + Sprachverarbeitung um CauseNet zu erweitern?
  + Einsatz von CNNs für Knowledge Graphs genauer untersuchen

# Stichpunkte zum Gespräch:

## Themen:

* 1)
* Machine learning
  + Stochastic gradient descent
* Optimizer, die sich nicht nur die erste oder zweite Ableitung ansieht
* Optimieren für knowledge graph embeddings
* 2)
* Hyperparameter optimieren
* Anzahl dimensionen
* Schichten neuronales Netz
* Meistens try and error
* Vielversprechendsten hyperparameter
* Bayeschen optimization
  + Wo ist die Unsicherheit am größten, diese stellen zu verbessern
* Meistens unvollständiges bild
* 3)
* Negative beispiele, KGE mit SPO triple
* Für neg beispiel wird objekt ausgetauscht
* Mit hoher wahrscheinlichkeit falsch 🡪 nicht unbedingt schlau
* Harte negative beispiele kreieren, die nah an pos beiliegt
* Wie berechnet man die ähnlichkeit von 2 Triplen?
* Muss schnell gehen, damit training nicht zu lange dauern
* Graphstatistiken anschauen?
* Pagerank für die Ähnlichkeit?
* .
* Generelle Vorgehensweise bei Abschlussarbeiten:
  + Benchmarking datensätze nutzen
    - FB15k-237: von Metaweb, von google aufgekauft
    - WN18RR: Wörter drin, die synonyme und wörter sind
    - YAGO3-10 etwas größerer datensatz für bspw Skalierbarkeit
  + 🡪KGE verfahren ausnutzen 3 Verfahren auf 2 Datensätzen beispielsweise:   
    TransE, DistMult, ComplEx, …, verfahren ausprobieren, optimierer austauschen
  + TransE, DistMult, .. gucken sich nur den graphen an 🡪 daher nichts mit NLP zu tun
  + Vll idee, auch NLP daten noch zu nutzen
* 4)  
  Typ von entitäten bestimmen
* 🡪connectE ansatz: berechnet 2 embeddings: für relationen, und einmal für die typrelation
* „Ist eine“ typbeziehung 🡪kombination dieser
* Weakly supervised learning 🡪nur für wenige traingsdaten lernen
* 2 neuronale netze
* Teacher student model -> auf die ungelabelten Daten -> wie das semi supervised learning
* 5)
* Use Contrastive learning for machine learning auf KG anpassen
* Time-series data, muss es nicht unbedingt sein
* Event ddurch graphen repräsentiert werden
* In dem Fall Bestellungen
* Übersicht in 🡪survey on contrastive self-supervised learning
* Idee:daten augmentiert, daten abgewandelt, aber noch positiv sein
* Normalerweise von Bildern

# Generelles zum Proposal:

* Proposal 2-3 Monate
* Proposal mit Ngonga besprechen
* Prototypen bereits mit Proposal bauen
* Schon zeitnah nach Proposal wird der Antrag eingereicht
* Proposal problem beschreiben, wie evaluieren, auf welchem Datensatz

# Generelles zur Abschlussarbeit:

* Früh anfangen zu schreiben
* Offizielle maximum 120 seiten
* Mind 60 seiten
* Parallel schreiben und implementieren
* Nicht erst programmierung fertigstellen
  + Wird dann knapp mit schreiben
* Implementierung zählt auch
* In erster linie wird text bewertet
* Am Ende 1 erste Version der Masterarbeit an Supervisor, vll auch unvollständiog um Feedback zu bekommen
* Auch nach ersten 1,2 Kapiteln fertiggestellt sind
  + Abhängig vom Supervisor und dem Studenten
* Meistens alle 2 Woche 1h treffen, aktuellen Stand besprechen
* Zum Antrag Abschlussarbeit in PAUL formular ausfüllen
* 2. Prüfer: Vorschlag von Ngonga, oder auch von mir

# TODO:

* Themen angucken, überlegen welche richtung
* Termin mit demir oder hamada
* Wenn ich vorher schon weiß, welche Themen nicht interessant/besonders interessant sind eine Email an Stefan schreiben
* **Termin: 2.9. 14 Uhr**
* Betreuer kommt dann dazu
* Themen für den Termin:
  + Themen aus ihrer Sicht vorstellen
  + Meine Fragen beantworten
  + Thema bereits genauer konkretisieren