**Folie 9: Lösungsansätze**

Jetzt kommen wir zu unseren Lösungsansätzen, die zum Ziel haben, bessere S2Match-Alignments zu finden. Gute Alignments sind der Schlüssel zum besseren, semantisch glaubwürdigen F-Score. Wir haben zwei Methoden in unserem Arsenal, mit denen wir das vor uns liegende Problem lösen.

**Folie 10: Methode 1 – Preprocessing AMRs**

Die erste Methode, die jetzt zum größten Teil fertig ist, macht im Prinzip einen Preprocessing-Schritt an den AMR-Graphen selbst und verändert sie in grundsätzlicher Weise. Die Idee ist, alle :mod-Relationen mit dem übergeordneten Knoten zu konkatenieren (An dem linken Graphen wäre der übergeordnete Knoten zum Beispiel r0 mit der Konstanten rover, da unter ihm ein paar :mod-Relationen stehen). Wir machen's aber ein bisschen komplexer, als einfach die Konstanten vom AMR-Graphen zu übernehmen (in diesem Fall country, moon, name, China). Der Grund dafür ist, dass die keine kohäsiven Phrasen bilden und das entsprechende Phrasen-Embedding semantisch nicht so gut wäre, da man in diesem Embedding AMR-Funktionswörter hätte (im Falle von Glove, wo wir einen Durchschnittsvektor bilden, wäre das vermutlich noch okay, aber wenn wir kontextuelle Embeddings haben, sollten wir schon kohäsive Phrasen haben).

Okay, jetzt könnte man natürlich fragen, warum wir ausgerechnet :mod-Relationen für diese Art von Konkatenation ausgewählt haben. Und die Antwort ist, dass sie für die meisten Adjunkte in Nominalphrasen zuständig sind (fast alle Attribute wie "a little", "chinese", "young" oder zusätzliche Substantive wie "key appeals court" werden in AMR durch :mod-Relationen dargestellt). Insofern ergab es aus unserer Sicht Sinn, uns speziell auf sie zu konzentrieren.

Es gibt ein paar ziemlich restriktive Voraussetzungen, wo solche Konkatenation durchgeführt wird, damit man nur die Fälle abdeckt, wo das tatsächlich was bringen könnte. Eine der Voraussetzungen ist, dass es keine Knoten im entsprechenden Teilbaum gibt, die außerhalb dieses Teilbaums erwähnt sind (also, es gibt Reentrancies, wie man sie im AMR-Jargon nennt). Das Problem, das wir in diesem Fall hätten ist, dass wir alle Knoten im Teilbaum zusammengeklappt haben, aber da gab es Variablen, die woanders verwendet wurden, die aber ihre Definitionen durch die Konkatenation verloren haben. Die dritte und meiner Meinung nach wichtigste Voraussetzung ist, dass wir diese Operation nur in den Fällen durchführen, wo der Teilbaum einem vollständigen Token-Span entspricht, das bedeutet, es gibt keine Lücken in AMR2Text-Alignments, wie man an dem Graphen rechts erkennen kann: "chinese lunar rover" ist ein vollständiges Span im Original-Satz, es gibt also keine Lücken. Auf den nächsten Folien habe ich ein paar Beispiele, wo eine der Voraussetzungen verletzt wird.

**Folie 11: Hier machen wir keine Transformation!**

Hier kann man sehen, dass keine Transformation stattgefunden hat, weil die Voraussetzung zu Reentrancies verletzt wurde: hier wird der Knoten Nummer 5 sowohl im Teilbaum von Knoten 3, als auch von Knoten 2 verwendet.

**Folie 12: Hier machen wir keine Transformation!**

Und an diesem Beispiel sieht man, dass sich durch Zusammenklappen von Knoten 1 ein unvollständiges Token-Span ergibt, es fehlt die Präposition und der Artikel vor "trilogy".

**Folie 13: Methode 2 – Postprocessing S2Match**

Die erste Methode, die ich grad vorgestellt habe, reduziert den AMR-Graphen und erleichtert somit das Auffinden von Embeddings-basierten Alignments in S2Match. An der vorgeschlagenen Methode kann man jedoch kritisieren, dass die Transformationen erstens nicht ganz übersichtlich sind, das heißt, wir können nur vermuten, wie der originale Graph vor der Transformation ausgesehen hat. Und zweitens sind solche Transformationen eigentlich nicht in allen Fällen nötig, sondern nur in den Fällen, wo für irgendwelche Knoten in einem Graphen keine Entsprechungen im zweiten Graphen gefunden wurden.

Deswegen arbeiten wir gerade an einer weiteren Methode, die einen Postprocessing-Schritt impliziert. Wir lassen also S2Match laufen und das Ergebnis sieht bsw. so aus, wie hier auf der linken Seite: hier sind die Knoten in grün markiert, für die ein Alignment gefunden wurde und rot markiert sind die Knoten, für die es keine Entsprechungen im zweiten Graphen gab.

Jetzt haben wir im Prinzip die gleiche Herangehensweise wie bei Methode 1: wir wollen Teilbäume als Phrasen darstellen und Phrasen als Vektoren. Doch hier setzen wir die :mod-Relation nicht voraus, sondern transformieren intern die Teilbäume, die einen ungemappten Knoten haben (außer natürlich der Wurzel, sonst haben wir keine AMR mehr nach der Transformation).

Methode 2 ist im Prinzip das gleiche wie Methode 1, aber hier beschränken wir uns nur auf die Fälle, wo es Probleme mit Alignierungen gab.