Jetzt kommen wir zu einem unserer Vorschläge, wie kompositionelle Ähnlichkeit erfasst werden kann. Wie wir schon aus den vorherigen Folien wissen, nimmt man bei S2Match Ähnlichkeiten von 2 lexikalischen Knoten in Betracht, was erlaubt, Similarität graduiert und nicht binär darzustellen. Was es aber noch nicht löst, ist, wie schon von meinen Kolleginnen erwähnt, wenn wir eine Phrase haben, die ein Konzept mit mehreren Wörtern ausdrückt. Ein Beispiel wären im Deutschen die sogenannten Funktionsverbgefüge, man kann sagen: "einen Antrag auf etwas stellen" oder einfach "beantragen", "Einfluss nehmen auf etwas" oder "beeinflussen".

In unseren Folien ist das Lieblingsbeispiel "a little cat" und "kitten". Sie sind natürlich alle nicht 100prozentig gleich, sind sich aber schon ähnlicher als "a little cat" und "a big car".) Rechts auf dieser Folie ist der AMR-Graph eines Satzes abgebildet: "In the story, evildoer Cruella de Vil makes no attempt to conceal her glee and greed." Uns kann an dieser Stelle ins Auge fallen, dass das Bild den Graphen in semantische Blöcke teilt. Wir sehen zum Beispiel in blau markiert den Namen des Bösewichtes – "Cruella de Vil", in grün macht der AMR eine Paraphrasierung von "evildoer", die einfach als "a person who does evil things" repräsentiert ist. Und diese Art von struktureller Paraphrasierung sehen wir eigentlich ganz oft in AMRs. Es benutzt sie, um komplexere Wörter und ihre Synonyme einheitlich zu repräsentieren, ohne ein großes Vokabular haben zu müssen. Nächste Folie bitte.

Meistens, abgesehen von Passivsätzen, ein paar Grenzfällen und extra Information in einem der Sätze, gehen wir davon aus, dass paraphrasierte Sätze aus semantisch ähnlichen Konstituenten oder Blöcken bestehen. Es wäre schön, wenn es eine Möglichkeit gäbe, die Konstituentenstruktur von AMR-Graphen direkt ablesen zu können. Das Problem ist nur, dass AMR sich nicht zum Ziel setzt, syntaktische Strukturen adäquat zu repräsentieren, Dependenzgraphen verfolgen dieses Ziel aber schon. Das Paper von Szubert beschäftigt sich eben mit der Aufgabe, syntaktische Blöcke, wie wir sie schon gesehen haben, in AMRs zu entdecken. Das Ziel dieses Papiers ist Dependenzstruktur eines Satzes auf seinen AMR-Graphen abzubilden. Somit haben wir eine Methode, die uns dabei hilft, AMRs in Blöcke zu teilen.

Okay, was wir jetzt damit machen können, ist folgendes: wir nehmen zwei Sätze, finden mit Hilfe eines AMR-Parsers und eines Dependenzparsers ihre Graphen: das heißt, ihre semantische, bzw. ihre syntaktische Struktur, finden für jeden Satz ein Mapping zwischen den beiden Strukturen. Für jeden syntaktischen Block können wir jetzt ein Embedding finden, zum Glück existiert eine auf Transformers basierte Bibliothek, die ganz gut mit Phrasenähnlichkeiten umgeht. Und schließlich ersetzen wir den lexikalischen Elternknoten eins AMR-Graphen durch ein Embedding, was sich danach in die schon existente Infrastruktur von S2Match integrieren lässt.

Gut, soviel zu unserem zweiten Vorschlag, jetzt kommen ein paar eher organisatorische Folien: es geht um eine genaue Auflistung von Tools und dem STS-Datensatz, den wir im Rahmen unserer Arbeit benutzen. Man kann dazu sagen, dass STS für semantic textual similarity steht und es geht vor allem darum, der Ähnlichkeit zweier Sätze eine Bewertung zu geben. Die Note 5 bekommen dabei die ähnlichsten Sätze, und 0 logischerweise die unähnlichsten. Ein Beispiel von einer 5 steht hier auf der linken Seite. Gut, jetzt kommt noch eine Folie mit unserem Zeitplan und Referenzen, die wir jetzt aber nicht unbedingt vorlesen müssen, denke ich. Danke für Eure Aufmerksamkeit!